



STIC Search Report

EIC 2100

STIC Database Tracking Number: 110252

TO: Hoang-Vu (Antony) Nguyen-Ba

Location:

Art Unit : 2122

Wednesday, December 17, 2003

Case Serial Number: 09184738

From: Terese Esterheld

Location: EIC 2100

PK2-4B30

Phone: 308-7795

Terese.esterheld@uspto.gov

Search Notes

Dear Examiner Nguyen-Ba,

Attached, please find the results of your search request for application 09184738. I have concentrated on finding information on the underlined topics on the request form.

The priority date of November 16, 1994 made it difficult to find items of value to you.

The enclosed set meets the requirement of the Priority date. Items have been marked, but there are probably additional items that are of value. Please look over the results carefully.

Please let me if you need additional information on this search.

Thank you for coming to EIC 2100.

Terese Esterheld





STIC EIC 2100 Search Request Form

110252
49

Today's Date:

12/11/03

What date would you like to use to limit the search?

Priority Date: 11/16/94

Other:

Name HOANG-VU "ANTONY" NGUYEN-BA

AU 2122 Examiner # 75507

Room # 5Y19 Phone 305-0103

Serial # 09/184,738

Format for Search Results (Circle One):

PAPER

DISK

EMAIL

Where have you searched so far?

USP

DWPI

EPO

JPO

ACM

IBM TDB

IEEE

INSPEC

SPI

Other

Is this a "Fast & Focused" Search Request? (Circle One) YES NO

A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at <http://ptoweb/patents/stic/stic-tc2100.htm>.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

The invention relates to a computer implemented system employing objects for generating an application script, in which the objects & the script may be maintained separately.

Applicant's system is comprising:
means for wrapping objects with additional properties & events beyond those properties & events internal to the object and
means for utilizing the additional properties & events to link & sequence the objects.

Applicant's definition of:

Object - modular s/w program written to a defined specification which may perform any function. For purposes of this application, the word "Object" shall be considered synonymous with the term "custom control". Objects are intended to be grouped & assembled into applications.

Script - a set of computer instructions written at a higher level than code, i.e., farther removed from machine language, which is usually interpreted during execution. "Script" for purposes of this application shall also refer to the listing of objects & property settings generated by the authoring program of this ~~program~~ invention.

Wrapping - process by which, when a new VBX control is added to the system →

STIC Searcher

Tereze Esterfeld

Phone

308-7795

Date picked up

12/15/03

4:45 pm

Date Completed

12/17/03

2:00 pm



Set	Items	Description
S1	26	AU='MORRIS R M'
S2	13	AU='MORRIS ROBERT' OR AU='MORRIS ROBERT M' OR AU='MORRIS R- OBERT MILTON'
S3	17	AU='DENTON L' OR AU='DENTON L E' OR AU='DENTON LEET E'
S4	51	S1 OR S2 OR S3
S5	7	S4 AND IC=G06F?

File 347:JAPIO Oct 1976-2003/Aug(Updated 031202)
(c) 2003 JPO & JAPIO

File 348:EUROPEAN PATENTS 1978-2003/Nov W05
(c) 2003 European Patent Office

File 349:PCT FULLTEXT 1979-2002/UB=20031211,UT=20031204
(c) 2003 WIPO/Univentio

File 350:Derwent WPIX 1963-2003/UD,UM &UP=200380
(c) 2003 Thomson Derwent

5/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2003 JPO & JAPIO. All rts. reserv.

07368292 **Image available**
SCHEDULING SYSTEM AND SCHEDULING METHOD

PUB. NO.: 2002-236789 [JP 2002236789 A]
PUBLISHED: August 23, 2002 (20020823)
INVENTOR(s): RASANSKY RICHARD A
DENTON LEET E
MORRIS ROBERT M
APPLICANT(s): TECHNO WING KK
APPL. NO.: 2001-359382 [JP 20011359382]
Division of 2000-528917 [JP 2000528917]
FILED: January 22, 1999 (19990122)
PRIORITY: 98 010963 [US 9810963], US (United States of America),
January 22, 1998 (19980122)
INTL CLASS: G06F-017/60

ABSTRACT

PROBLEM TO BE SOLVED: To provide a computer system enabling an end user to communicate and confirm with this system by including an end user interface to be sent by using a standard communication protocol.

SOLUTION: In the computer system for scheduling an event between the end users of the system, each end users 20 are given is granted a unique password to protect a personal calendar. This calendar is generated from information stored in a database at a central server 10, and delivered to each end user 20 as a standard HTML sent through the Internet 30. This custom personal calendar can be viewed in a standard Web browser by the end users 20.

COPYRIGHT: (C)2002,JPO

5/5/2 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

01075348

SCHEDULING SYSTEM
ZEIT-PLANUNGS-SYSTEM
SYSTEME D'ETABLISSEMENT DU CALENDRIER
PATENT ASSIGNEE:

Appointnet, Inc., (2815570), 234 N. Columbus Boulevard, Philadelphia, PA 19106, (US), (Applicant designated States: all)

INVENTOR:

RASANSKY, Richard, A., 222 South Van Pelt Avenue, Philadelphia, PA 19103, (US)

DENTON, Leet, E. , 148 Stoneway Lane, Bala Cynwyd, PA 19004, (US)

MORRIS, Robert, M. , P.O. Box 105, Westtown, PA 19395, (US)

LEGAL REPRESENTATIVE:

Allan, James Stewart et al (90951), Murgitroyd & Company, 373 Scotland Street, Glasgow G5 8QA, (GB)

PATENT (CC, No, Kind, Date): EP 1049983 A1 001108 (Basic)
WO 9938079 990729

APPLICATION (CC, No, Date): EP 99902406 990122; WO 99US1239 990122

PRIORITY (CC, No, Date): US 10963 980122

DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE

EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI

INTERNATIONAL PATENT CLASS: G06F-013/00

NOTE:

No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application: 001108 A1 Published application with search report

Application: 990929 A1 International application. (Art. 158(1))
Withdrawal: 030514 A1 Date application deemed withdrawn: 20020801
Examination: 001108 A1 Date of request for examination: 20000731
Application: 990929 A1 International application entering European
phase

LANGUAGE (Publication,Procedural,Application): English; English; English

5/5/3 (Item 1 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2003 WIPO/Univentio. All rts. reserv.

00506727 **Image available**

SCHEDULING SYSTEM

SYSTEME D'ETABLISSEMENT DU CALENDRIER

Patent Applicant/Assignee:

APPOINTNET INC,
RASANSKY Richard A,
DENTON Leet E,
MORRIS Robert M,

Inventor(s):

RASANSKY Richard A,
DENTON Leet E ,
MORRIS Robert M

Patent and Priority Information (Country, Number, Date):

Patent: WO 9938079 A1 19990729
Application: WO 99US1239 19990122 (PCT/WO US9901239)
Priority Application: US 9810963 19980122

Designated States: AU BB BG BR CA CN CU CZ EE GE HU ID IL IS JP KP KR LC LK
LR LT LV MG MK MN MX NO NZ PL RO SG SI SK TR TT UA UG US UZ VN YU ZW GH
GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES
FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN
TD TG

Main International Patent Class: G06F-013/00

Publication Language: English

Fulltext Availability:

Detailed Description
Claims

Fulltext Word Count: 13970

English Abstract

A computer system for scheduling events between end users (20) of the system. Each end user (20) is granted a unique password protected personal calendar. This calendar is generated from information stored in a database at a central server (10), and delivered to each end user (20) as standard HTML sent through the Internet (30). This custom personal calendar is then viewed by the end user (20) in a standard Web Browser. This obviates the need for special software programs to be purchased by end users, and also allows end users of any CPU type to read their calendars. When an end user (20) uses the system to send an Invitation or Announcement to others on the system, the sending user (20) has the option of sending E-mail in addition to posting that information in the calendars' of others. When an end user (20) sends an Invitation or Announcement to a person who is not an Appointnet user, then the Appointnet system automatically creates a unique calendar for the recipient, and sends an E-mail to that person. Individuals who use the present system can post reminders to themselves, send announcements to people they know, and make appointments with people they know. When these messages are sent, the communication is nearly instantaneous because the system makes one record and allows both (or many) parties to view it.

French Abstract

L'invention concerne un systeme informatique d'etablissement du calendrier entre utilisateurs (20) du systeme. Chaque utilisateur (20) recoit un calendrier personnel protege par un mot de passe unique. Ce calendrier est etabli d'apres les informations d'une base de donnees propre a un serveur central (10), et la fourniture a chaque utilisateur

(20) s'effectue en format HTML classique via Internet (30). Ledit calendrier personnel est ensuite consulte par l'utilisateur (20) via un explorateur Web classique. Cela evite l'achat de programmes logiciels speciaux et permet en outre a l'utilisateur d'utiliser une unite centrale quelconque pour consulter le calendrier. Lorsqu'un utilisateur (20) fait appel au systeme pour envoyer une invitation ou une annonce a d'autres utilisateurs, cet expediteur (20) peut envoyer un courrier electronique en plus de l'information consideree destinee aux calendriers de destination. Lorsqu'un utilisateur (20) envoie une invitation ou une annonce a des utilisateurs qui ne font pas partie d'Appointnet, le systeme Appointnet cree automatiquement un calendrier unique pour le destinataire, et il envoie un courrier electronique a la personne. Les individus qui utilisent le systeme peuvent s'adresser personnellement des rappels, envoyer des annonces a des connaissances et fixer des rendez-vous a ces connaissances. Une fois les messages envoyes, la communication est presque instantanee parce que le systeme etablit un dossier et permet aux deux (nombreuses) parties de le consulter.

5/5/4 (Item 1 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014978737 **Image available**

WPI Acc No: 2003-039251/200303

Related WPI Acc No: 2003-039250

XRPX Acc No: N03-030637

Streaming video delivery control method for satellite video delivery networks, involves forming user interface within web page hosted by web browser that is superimposed over video picture

Patent Assignee: GREENWALT T (GREE-I); MORRIS R M (MORR-I)

Inventor: GREENWALT T; MORRIS R M

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020116708	A1	20020822	US 2000254708	A	20001211	200303 B
			US 2001956492	A	20010918	

Priority Applications (No Type Date): US 2000254708 P 20001211; US 2001956492 A 20010918

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020116708	A1		10	G06F-003/00	Provisional application US 2000254708

Abstract (Basic): US 20020116708 A1

NOVELTY - A web browser (302) is superimposed over a video picture displayed after decoding a corresponding video stream. A user interface is formed within a web page hosted by the web browser by generating browser controls in the web page to form the user interface.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

- (1) User interface provision method in a digital media content receiver system;
- (2) Streaming video delivery controlling system;
- (3) User interface for controlling streaming media content; and
- (4) User interface formation method.

USE - For cable or satellite video delivery networks.

ADVANTAGE - Provides the user interface as a set of browser-based control functions and is hosted by the web browser provided by the media gateway, thus providing browser-in-browser functionality and the component used can be re-used in several applications. The user interface provides the ability to customize the streaming media client, so that several functions can be provided to the users.

DESCRIPTION OF DRAWING(S) - The figure shows an overlay mixer and filter for generating the user interface.

Web browser (302)

pp; 10 DwgNo 3/3

Title Terms: STREAM; VIDEO; DELIVER; CONTROL; METHOD; SATELLITE; VIDEO;

DELIVER; NETWORK; FORMING; USER; INTERFACE; WEB; PAGE; WEB; SUPERIMPOSED;
VIDEO; PICTURE
Derwent Class: P85; T01; W02; W03
International Patent Class (Main): G06F-003/00
International Patent Class (Additional): G06F-013/00 ; G09G-005/00;
H04N-005/445; H04N-007/173
File Segment: EPI; EngPI

5/5/5 (Item 2 from file: 350)
DIALOG(R) File 350:Derwent WPIX
(c) 2003 Thomson Derwent. All rts. reserv.

014978736 **Image available**
WPI Acc No: 2003-039250/200303
Related WPI Acc No: 2003-039251
XRPX Acc No: N03-030636

Client system for real-time digital media service provision, has media control interface for transmitting control signal to digital media engine based on signals received corresponding to control program in web page selected by user

Patent Assignee: GREENWALT T (GREE-I); MORRIS R M (MORR-I)

Inventor: GREENWALT T; MORRIS R M

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020116707	A1	20020822	US 2000254708	A	20001211	200303 B
			US 2001956481	A	20010918	

Priority Applications (No Type Date): US 2000254708 P 20001211; US 2001956481 A 20010918

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020116707	A1		10	H04N-007/08	Provisional application US 2000254708

Abstract (Basic): US 20020116707 A1

NOVELTY - A digital media engine (110) receives and processes a digital content. A media gateway (112) hosts user interface web page including user-selectable control programs for controlling the digital media engine. A media control interface (109) transmits control signals to the digital media engine according to the signals received corresponding to the control program selected by user.

USE - For real-time digital media service such as satellite television broadcast, movies-on-demand service, multimedia service. etc.

ADVANTAGE - Provides high-quality multimedia services in real-time without degrading artifacts and reduces total time to download and process a file.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of client system.

Media control interface (109)
Digital media engine (110)
Media gateway (112)
pp; 10 DwgNo 1/3

Title Terms: CLIENT; SYSTEM; REAL; TIME; DIGITAL; MEDIUM; SERVICE; PROVISION; MEDIUM; CONTROL; INTERFACE; TRANSMIT; CONTROL; SIGNAL; DIGITAL; MEDIUM; ENGINE; BASED; SIGNAL; RECEIVE; CORRESPOND; CONTROL; PROGRAM; WEB; PAGE; SELECT; USER

Derwent Class: P85; T01; W02; W03
International Patent Class (Main): H04N-007/08
International Patent Class (Additional): G06F-003/00 ; G06F-013/00 ; G09G-005/00; H04N-005/445; H04N-007/84; H04N-007/87
File Segment: EPI; EngPI

5/5/6 (Item 3 from file: 350)
DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

012687868 **Image available**

WPI Acc No: 1999-493977/199941

XRPX Acc No: N99-367973

Computer system for scheduling events e.g. meetings, appointments, announcements

Patent Assignee: APPOINTNET INC (APPO-N); ECAL CORP (ECAL-N); TECHNO WING KK (TECH-N)

Inventor: DENTON L E ; MORRIS R M ; RASANSKY R A

Number of Countries: 071 Number of Patents: 008

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9938079	A1	19990729	WO 99US1239	A	19990122	199941 B
US 5960406	A	19990928	US 9810963	A	19980122	199947
AU 9922392	A	19990809	AU 9922392	A	19990122	200001
EP 1049983	A1	20001108	EP 99902406	A	19990122	200062
			WO 99US1239	A	19990122	
JP 2002501249	W	20020115	WO 99US1239	A	19990122	200207
			JP 2000528917	A	19990122	
CN 1330784	A	20020109	CN 99804354	A	19990122	200229
JP 3294840	B2	20020624	WO 99US1239	A	19990122	200243
			JP 2000528917	A	19990122	
JP 2002236789	A	20020823	JP 2000528917	A	19990122	200271
			JP 2001359382	A	19990122	

Priority Applications (No Type Date): US 9810963 A 19980122

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 9938079 A1 E 82 G06F-013/00

Designated States (National): AU BB BG BR CA CN CU CZ EE GE HU ID IL IS JP KP KR LC LK LR LT LV MG MK MN MX NO NZ PL RO SG SI SK TR TT UA UG US UZ VN YU ZW

Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW

US 5960406 A G06F-017/60

AU 9922392 A G06F-013/00 Based on patent WO 9938079

EP 1049983 A1 E G06F-013/00 Based on patent WO 9938079

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI

JP 2002501249 W 153 G06F-017/60 Based on patent WO 9938079

CN 1330784 A G06F-013/00

JP 3294840 B2 61 G06F-017/60 Previous Publ. patent JP 200201249

Based on patent WO 9938079

JP 2002236789 A 42 G06F-017/60 Div ex application JP 2000528917

Abstract (Basic): WO 9938079 A1

NOVELTY - The computer system comprises an end user interface that is generated in a standard page markup syntax language and which is accessible to at least two end users (20) of the system using standard communication protocols, to allow one of the at least two end users to establish and send an event to be scheduled within the system to the second end user.

DETAILED DESCRIPTION - Each end user is granted a unique password-protected personal calendar. The calendar is generated from information stored in a database at a central server (10), and delivered to each end user as standard HTML sent through the Internet (30). The custom personal calendar is viewed by the end user in a standard Web Browser. When an end user uses the system to send an invitation or announcement to others on the system, the sending user has the option of sending E-mail in addition to posting the information in the calendars of the others. When an end user sends an invitation or announcement to a person who is not a system user, the system automatically creates a unique calendar for the recipient and sends an E-mail to that person.

An INDEPENDENT CLAIM is included for a method for scheduling events.

USE - For scheduling events e.g. meetings, appointments, announcements.

ADVANTAGE - Provides a simple and efficient way for people to make appointments, arrange meetings, and create other sorts of events that occur at an agreed time. Does not require end users to purchase any special software program since the custom personal calendar is viewed by the end user in a standard Web Browser. Allows end users of any CPU type to read their calendars.

DESCRIPTION OF DRAWING(S) - The figure is a system diagram of the scheduling system.

Central server (10)

End users (20)

Internet (30)

pp; 82 DwgNo 1/27

Title Terms: COMPUTER; SYSTEM; SCHEDULE; EVENT; ANNOUNCE

Derwent Class: T01; W01

International Patent Class (Main): G06F-013/00 ; G06F-017/60

File Segment: EPI

5/5/7 (Item 4 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

012324642 **Image available**

WPI Acc No: 1999-130748/199911

XRPX Acc No: N99-095179

Computer implemented application development system - sets properties of additional objects until all of desired objects have been specified, and then interconnecting objects in temporal sequences

Patent Assignee: DENTON L E (DENT-I); MORRIS R M (MORR-I)

Inventor: DENTON L E ; MORRIS R M

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5862372	A	19990119	US 94340702	A	19941116	199911 B

Priority Applications (No Type Date): US 94340702 A 19941116

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5862372	A		11	G06F-009/06	

Abstract (Basic): US 5862372 A

NOVELTY - The objects written at standard specification are wrapped. Four views are established and synchronized. The object is moved into one of the four views, and properties are set to the object. The properties are set to additional objects until all the desired objects have been specified. The objects are interconnected in temporal sequences. The flow of data and control between objects is specified. A script reflector is generated and the script is executed.

USE - For authoring application system.

ADVANTAGE - Icons representing the objects are placed into appropriate view, even though user does not know how to write specialized code. DESCRIPTION OF DRAWING(S) - The figure shows simple program for four views, output, map, multitrack and work form simultaneously displayed in separate windows.

Dwg.5/6

Title Terms: COMPUTER; IMPLEMENT; APPLY; DEVELOP; SYSTEM; SET; PROPERTIES;

ADD; OBJECT; OBJECT; SPECIFIED; INTERCONNECT; OBJECT; TEMPORAL; SEQUENCE

Derwent Class: T01

International Patent Class (Main): G06F-009/06

International Patent Class (Additional): G06F-009/22

File Segment: EPI

Set	Items	Description
S1	10243	AU=(MORRIS, R? OR MORRIS R? OR DENTON, L? OR DENTON L?)
S2	1	S1 AND APPLICATION()DEVELOPMENT
S3	1	S1 AND STANDARD?()OBJECT?
S4	2	S2 OR S3
File	2:INSPEC 1969-2003/Dec W1	(c) 2003 Institution of Electrical Engineers
File	6:NTIS 1964-2003/Dec W2	(c) 2003 NTIS, Intl Cpyrghrt All Rights Res
File	8:Ei Compendex(R) 1970-2003/Dec W1	(c) 2003 Elsevier Eng. Info. Inc.
File	34:SciSearch(R) Cited Ref Sci 1990-2003/Dec W1	(c) 2003 Inst for Sci Info
File	35:Dissertation Abs Online 1861-2003/Nov	(c) 2003 ProQuest Info&Learning
File	65:Inside Conferences 1993-2003/Dec W2	(c) 2003 BLDSC all rts. reserv.
File	92:IHS Intl.Stds.& Specs. 1999/Nov	(c) 1999 Information Handling Services
File	94:JICST-EPlus 1985-2003/Dec W2	(c)2003 Japan Science and Tech Corp(JST)
File	95:TEME-Technology & Management 1989-2003/Nov W5	(c) 2003 FIZ TECHNIK
File	99:Wilson Appl. Sci & Tech Abs 1983-2003/Oct	(c) 2003 The HW Wilson Co.
File	103:Energy SciTec 1974-2003/Nov B2	(c) 2003 Contains copyrighted material
File	144:Pascal 1973-2003/Dec W1	(c) 2003 INIST/CNRS
File	202:Info. Sci. & Tech. Abs. 1966-2003/Nov 17	(c) 2003 EBSCO Publishing
File	233:Internet & Personal Comp. Abs. 1981-2003/Jul	(c) 2003, EBSCO Pub.
File	239:Mathsci 1940-2003/Jan	(c) 2003 American Mathematical Society
File	275:Gale Group Computer DB(TM) 1983-2003/Dec 12	(c) 2003 The Gale Group
File	434:SciSearch(R) Cited Ref Sci 1974-1989/Dec	(c) 1998 Inst for Sci Info
File	647:CMP Computer Fulltext 1988-2003/Dec W2	(c) 2003 CMP Media, LLC
File	674:Computer News Fulltext 1989-2003/Dec W1	(c) 2003 IDG Communications
File	696:DIALOG Telecom. Newsletters 1995-2003/Dec 15	(c) 2003 The Dialog Corp.

4/5,K/1 (Item 1 from file: 2)

DIALOG(R) File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

5946476 INSPEC Abstract Number: C9807-6160J-011

Title: A temporal object query language

Author(s): Fegaras, L.; Elmasri, R.

Author Affiliation: Dept. of Comput. Sci. & Eng., Texas Univ., Arlington, TX, USA

Conference Title: Proceedings. Fifth International Workshop on Temporal Representation and Reasoning (Cat. No.98EX157) p.51-7

Editor(s): Khatib, L.; Morris, R.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA

Publication Date: 1998 Country of Publication: USA ix+191 pp.

ISBN: 0 8186 8473 9 Material Identity Number: XX98-01337

U.S. Copyright Clearance Center Code: 0 8186 8473 9/98/\$10.00

Conference Title: Proceedings. Fifth International Workshop on Temporal Representation and Reasoning (Cat. No.98EX157)

Conference Sponsor: Florida Artificial Intelligence Res. Soc. (FLAIRS)

Conference Date: 16-17 May 1998 Conference Location: Sanibel Island, FL, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: One of the main reasons that current commercial DBMSs provide only minimal support for temporal concepts is the size and complexity involved. Object-oriented databases, on the other hand, were developed to deal with complex database applications. Because temporal concepts require complex type support and advanced modeling concepts, object-oriented databases are excellent candidates for realization of temporal databases without requiring fundamental extensions to the basic data model. The ODMG (Object Data Management Group) has proposed a standard for object-oriented databases, including a **standard object** model, an object query language (OQL), and an object definition language (ODL). These do not include temporal support except at the data type level, as in SQL2. We present a language extension to OQL to accommodate time information. Our goal is not to propose a new temporal query language, but to incorporate temporal features into the existing OQL framework. (14 Refs)

Subfile: C

Descriptors: object-oriented databases; query languages; temporal databases

Identifiers: temporal object query language; commercial DBMSs; temporal concepts; object-oriented databases; complex type support; advanced modeling concepts; temporal databases; data model; ODMG; **standard object** model; object query language; OQL; object definition language; ODL; time information

Class Codes: C6160J (Object-oriented databases); C6160Z (Other DBMS); C6140D (High level languages)

Copyright 1998, IEE

...Abstract: model. The ODMG (Object Data Management Group) has proposed a standard for object-oriented databases, including a **standard object** model, an object query language (OQL), and an object definition language (ODL). These do not include temporal...

...Identifiers: **standard object** model

Khatib, L. (editor); Morris, R. (editor)

4/5,K/2 (Item 1 from file: 275)

DIALOG(R) File 275:Gale Group Computer DB(TM)

(c) 2003 The Gale Group. All rts. reserv.

01285087 SUPPLIER NUMBER: 07264047 (USE FORMAT 7 OR 9 FOR FULL TEXT)

A graphic comparison; Unix versus OS-2.

Morris, Robert R. ; Brooks, William E

PC Tech Journal, v7, n2, p106(9)

Feb, 1989

DOCUMENT TYPE: evaluation

ISSN: 0738-0194

LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6663 LINE COUNT: 00543

ABSTRACT: The OS-2 Presentation Manager and the Unix X-Window graphical user-interfaces are compared. Presentation Manager provides a consistent user interface, an extensive programming interface and device-independent graphics for a single workstation or microcomputer. X Window is available in a variety of forms, has a skeletal programming interface, and provides a distributed network implementation. Presentation Manager is the more fully-realized of the two, but it is built on an unproven operating system. X Window may benefit from extensions such as AT and T and Sun Microsystems Inc's Open Look or DEC's DECwindows, both of which include more complete sets of widgets (graphics control elements such as buttons), standards for using them and richer sets of application programming interfaces (APIs).

CAPTIONS: Event messages. (table); Resource-building facilities. (table)

SPECIAL FEATURES: illustration; table

DESCRIPTORS: UNIX; GUI; Evaluation; Comparison; Computer Graphics

SIC CODES: 7372 Prepackaged software

TRADE NAMES: X Window System (Program development software)--evaluation;

OS/2 Presentation Manager (GUI)--evaluation

OPERATING PLATFORM: Unix; OS-2

FILE SEGMENT: CD File 275

Morris, Robert R ...

... version then in use. The X.11 release was so incompatible with X.10.4 that most **application development** was curtailed until X.11 became available. Release X.11.2 was introduced in March 1988, when...

Set	Items	Description
S1	640971	OBJECT? OR CUSTOM(1W)CONTROL? OR SOFTWARE()ROUTINE? OR DLL OR DLLS OR CODE()MODULE? OR VBX OR VBXS OR VISUAL()BASIC()EXT- ENSION? OR READY()MADE() (COMPONENT? OR CONTROL?) OR OCX OR OC- XS OR (ACTIVEX OR ACTIVE()X)()CONTROL?
S2	6084621	GENERAT? OR MAKE? ? OR MAKING OR CREATE? OR CREATING OR PR- ODUC?
S3	474583	SCRIPT? OR INSTRUCTION? OR RULE? OR SYNTAX OR CODE OR CODES
S4	7546	(MAINTAIN? OR PRESERVE? OR KEEP? ? OR SUSTAIN? OR SUPPORT?-) (3N) (SEPARATELY OR INDEPENDENTLY OR INDIVIDUALLY OR SINGLY)
S5	6537620	WRAP? OR CONTAIN? OR INCLUDE? OR HOLD? OR ENCLOSE?
S6	6736069	ADDITIONAL OR ANOTHER OR SECOND OR 2ND OR DIFFERENT OR MORE OR SECONDARY OR SUPPLEMENT? OR OTHER
S7	2540374	PROPERTY OR PROPERTIES OR FIELD? OR PATTERN? OR QUALITY OR QUALITIES OR ATTRIBUTE? OR FEATURE?
S8	518795	EVENT? OR HAPPENING? OR INCIDENT? OR OCCURRENCE? OR DEVELO- PMENT? OR EPISODE?
S9	1277931	ABOVE OR BEYOND OR EXCEED? OR SURPASS? OR OVER()REACH? OR - OVERREACH?
S10	4855779	INTERNAL OR CONFIGUR? OR DESIGN? OR ARRANGE? OR SET()UP OR SETUP OR PROGRAM?()IN OR SETTING? OR IMBED? OR EMBED? OR FIXED OR ENTRENCH? OR FASTEN? OR INFIX? OR INGRAIN
S11	7357961	UTILIZE OR APPLY? OR APPLIES OR EMPLOY? OR IMPLEMENT? OR U- SE OR USES?
S12	5228459	LINK? OR (PUT OR LAY OR PIECE)()TOGETHER OR JOIN? OR UNITE? OR UNIFY OR UNIFIES OR CONNECT? OR ASSEMBLE? OR COLLECT? OR - COMBINE? OR ASSOCIAT? OR AFFILIAT? OR TIE? OR TRANSACTION?
S13	425955	SEQUENCE? OR QUEUE OR QUEUED OR QUEUING OR QUEUEING OR RAN- K? OR (LOGICAL OR SERIAL)()ORDER OR CONSECUTIVE?
S14	12732	S1 AND S2 AND S3
S15	3380	S5 AND S1 AND (S6 (5N) (S7 OR S8))
S16	38591	(S7 OR S8) AND S10 AND S1
S17	79669	S11 AND (S6 (5N) (S7 OR S8))
S18	154039	(S12 OR S13) AND S1
S19	5	S14 AND S4
S20	46	S14 AND S15 AND S16 AND S1 AND S18
S21	66	S14 AND S15 AND S16
S22	146	S14 AND S15 AND S17
S23	81	S22 AND S18
S24	38	S14 AND S15 AND S16 AND S17 AND S18
S25	114	S19 OR S20 OR S21 OR S23 OR S24
S26	68	S25 AND IC=G06F?
S27	28	S26 AND IC=G06F-009?
S28	8	S25 AND MC=(T01-F01 OR T01-F06 OR T01-J12 OR T01-J20B1)
S29	34	S27 OR S28

File 347:JAPIO Oct 1976-2003/Aug(Updated 031202)

(c) 2003 JPO & JAPIO

File 350:Derwent WPIX 1963-2003/UD,UM &UP=200381

(c) 2003 Thomson Derwent

29/5/2 (Item 2 from file: 347)
DIALOG(R) File 347:JAPIO
(c) 2003 JPO & JAPIO. All rts. reserv.

03248527 **Image available**
FAULT RECOVERY SUPPORT SYSTEM

PUB. NO.: 02-224027 [JP 2224027 A]
PUBLISHED: September 06, 1990 (19900906)
INVENTOR(s): KAWASHIMA HARUHITO
YASUNOBU SEIJI
MONNO TAKATOMO
TAKIGUCHI HITOSHI
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP
(Japan)
APPL. NO.: 01-042967 [JP 8942967]
FILED: February 27, 1989 (19890227)
INTL CLASS: [5] G06F-009/44
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1134, Vol. 14, No. 530, Pg. 55,
November 21, 1990 (19901121)

ABSTRACT

PURPOSE: To improve the performance regarding maintenance by performing a reasoning process by an inference mechanism by referring to the knowledge in a display and inference **object** knowledge base consisting of an intermediate knowledge base, and displaying information.

CONSTITUTION: An intermediate knowledge base constituting mechanism 6 constitutes the intermediate knowledge base 7 composed of a **rule** group indicating a recovery procedure corresponding to input fault items in a tree structure description **rule** type knowledge base 4 and display information referred to in recovery procedure **rules** according to both the pieces of information. A display and inference knowledge base constituting mechanism 8 converts the **generated** intermediate knowledge 7 into a display and inference knowledge base 10 to be executed by the inference mechanism for displaying information and judging a branch. Then a display and inference mechanism 9 refers to the knowledge in the display and inference knowledge base 10 to display the recovery procedure through a man-machine interface 12 and also judges the branch according to the answer input of a process result. Consequently, the **rule** type knowledge regarding the fault recovery procedure and display information to be displayed by the respective procedures are **maintained independently**.

29/5/22 (Item 20 from file: 350)
DIALOG(R) File 350:Derwent WPIX
(c) 2003 Thomson Derwent. All rts. reserv.

012496043 **Image available**
WPI Acc No: 1999-302151/199925
Related WPI Acc No: 1998-193113; 1998-446392; 1998-456625
XRPX Acc No: N99-226375

Object oriented computer user interface for computer system

Patent Assignee: MASSACHUSETTS INST TECHNOLOGY (MASI)
Inventor: BERENSON R W; LAI K; MALONE T W; YU K
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5900870	A	19990504	US 89375416	A	19890630	199925 B
			US 92888249	A	19920522	
			US 94336712	A	19941109	

Priority Applications (No Type Date): US 89375416 A 19890630; US 92888249 A
19920522; US 94336712 A 19941109

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

US 5900870 A 52 G06F-015/00 Cont of application US 89375416
Cont of application US 92888249

Abstract (Basic): US 5900870 A

NOVELTY - A display unit provided in graphical representation unit displays summary of **collection** of **objects** showing one or **more fields** selected by user from each of the **objects**. The graphical representation unit and editor permits action upon editing of each **object** by user in substantially uniform way regardless of type.

DETAILED DESCRIPTION - The **objects** that are instances of several **object** types **collectively** are graphically represented by a graphical representation unit, where the **object** has several fields atleast one among which **contains** information of any type. The **object** types are **created** and edited by an editor. Each instance of **object** type is **created**, edited and displayed using a template. An INDEPENDENT CLAIM is also **included** for the computer system.

USE - For computer system.

ADVANTAGE - Enables multiple user to access and modify contents of database in uniform, intuitive and simple manner.

DESCRIPTION OF DRAWING(S) - The figure shows instance of **rule**.

pp; 52 DwgNo 5/26

Title Terms: **OBJECT**; **ORIENT**; **COMPUTER**; **USER**; **INTERFACE**; **COMPUTER**; **SYSTEM**

Derwent Class: T01

International Patent Class (Main): G06F-015/00

File Segment: EPI

29/5/23 (Item 21 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

012324642 **Image available**

WPI Acc No: 1999-130748/199911

XRPX Acc No: N99-095179

Computer implemented **application development system** - sets properties of additional **objects** until all of desired **objects** have been specified, and then interconnecting **objects** in temporal sequences

Patent Assignee: DENTON L E (DENT-I); MORRIS R M (MORR-I)

Inventor: DENTON L E; MORRIS R M

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5862372	A	19990119	US 94340702	A	19941116	199911 B

Priority Applications (No Type Date): US 94340702 A 19941116

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 5862372	A	11	G06F-009/06	

Abstract (Basic): US 5862372 A

NOVELTY - The **objects** written at standard specification are **wrapped**. Four views are established and synchronized. The **object** is moved into one of the four views, and properties are set to the **object**. The **properties** are set to **additional objects** until all the desired **objects** have been specified. The **objects** are interconnected in temporal **sequences**. The flow of data and control between **objects** is specified. A **script** reflector is **generated** and the **script** is executed.

USE - For authoring application system.

ADVANTAGE - Icons representing the **objects** are placed into appropriate view, even though user does not know how to write specialized **code**. DESCRIPTION OF DRAWING(S) - The figure shows simple program for four views, output, map, multitrack and work form simultaneously displayed in separate windows.

Dwg.5/6

Title Terms: **COMPUTER**; **IMPLEMENT**; **APPLY**; **DEVELOP**; **SYSTEM**; **SET**;

PROPERTIES; **ADD**; **OBJECT**; **OBJECT**; **SPECIFIED**; **INTERCONNECT**; **OBJECT**;
TEMPORAL; **SEQUENCE**

*Derwent Class: T01
International Patent Class (Main): G06F-009/06
International Patent Class (Additional): G06F-009/22
File Segment: EPI

29/5/27 (Item 25 from file: 350)
DIALOG(R) File 350:Derwent WPIX
(c) 2003 Thomson Derwent. All rts. reserv.

010620378 **Image available**
WPI Acc No: 1996-117331/199613
XRPX Acc No: N96-098116

Constructing display of partially-ordered data e.g. for debugging of
concurrent execution program e.g. distributed system - determining
events for which no other event precedes them, displaying events
in farthest position consistent with partial order, and repositioning
events if operation direction is opposite traversal direction

Patent Assignee: TAYLOR D J (TAYL-I); INT BUSINESS MACHINES CORP (IBM C)
Inventor: TAYLOR D J
Number of Countries: 007 Number of Patents: 009
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
AU 9520127	A	19960208	AU 9520127	A	19950516	199613 B
JP 8055044	A	19960227	JP 95158917	A	19950626	199618
EP 706162	A1	19960410	EP 95305283	A	19950728	199619
CA 2129190	A	19960130	CA 2129190	A	19940729	199620
US 5604851	A	19970218	US 95449854	A	19950524	199713
US 5640500	A	19970617	US 95449854	A	19950524	199730
			US 95469320	A	19950602	
CA 2129190	C	19990817	CA 2129190	A	19940729	199953
EP 706162	B1	20030917	EP 95305283	A	19950728	200369
DE 69531759	E	20031023	DE 631759	A	19950728	200377
			EP 95305283	A	19950728	

Priority Applications (No Type Date): CA 2129190 A 19940729

Cited Patents: 02Jnl.Ref

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
AU 9520127	A		59	G06F-003/14	
JP 8055044	A		25	G06F-011/28	
EP 706162	A1 E		32	G09G-001/00	
Designated States (Regional): DE FR GB					
CA 2129190	A			G09F-009/00	
US 5604851	A		29	G06F-015/00	
US 5640500	A		26	G06F-015/00	Div ex application US 95449854
CA 2129190	C E			G06F-003/14	
EP 706162	B1 E			G09G-001/00	
Designated States (Regional): DE FR GB					
DE 69531759	E			G09G-001/00	Based on patent EP 706162

Abstract (Basic): AU 9520127 A

The method provides displays such that the information is
maintained in the partial order while allowing intuitive scrolling of
the information on the display through the
implementation of predefined unidirectional movement,
monotonicity, priority, maximal display and minimal change constraints.

The arrangement of the information on the display may be changed
by the user and appropriate labels and identifiers can be included as
desired.

USE - Useful event tracer for transaction processing for
database and object processing in object-oriented programming.
Information may also be grouped into clusters to reduce clutter on
display, allowing information of particular interest to be examined
more easily.

Dwg.8/18

Title Terms: CONSTRUCTION; DISPLAY; ORDER; DATA; DEBUG; CONCURRENT; EXECUTE
; PROGRAM; DISTRIBUTE; SYSTEM; DETERMINE; **EVENT** ; NO; **EVENT** ; PRECEDE;
DISPLAY; **EVENT** ; POSITION; CONSISTENT; ORDER; REPOSITION; **EVENT** ;
OPERATE; DIRECTION; OPPOSED; TRAVERSE; DIRECTION
Derwent Class: P85; T01; W04
International Patent Class (Main): **G06F-003/14** ; **G06F-011/28** ;
G06F-015/00 ; G09F-009/00; G09G-001/00
International Patent Class (Additional): G01R-031/318; **G06F-009/06** ;
G06F-011/00 ; **G06F-011/32** ; **G06F-011/34** ; **G06F-017/00** ; G09G-005/00;
H04N-005/262
File Segment: EPI; EngPI

29/5/28 (Item 26 from file: 350)
DIALOG(R) File 350:Derwent WPIX
(c) 2003 Thomson Derwent. All rts. reserv.

010402387 **Image available**
WPI Acc No: 1995-303700/199540
XRPX Acc No: N95-230718

**Interactive information processing system with agent capability -
generates common platform composed of multiple windows and converts data
between visible object rendered in common platform and data structure
discernible by computer**

Patent Assignee: FUJITSU LTD (FUIT)
Inventor: ENOMOTO H
Number of Countries: 005 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 670544	A2	19950906	EP 95103004	A	19950302	199540 B
CA 2143914	A	19950905	CA 2143914	A	19950303	199549
JP 7295929	A	19951110	JP 9541572	A	19950301	199603
EP 670544	A3	19960710	EP 95103004	A	19950302	199636
US 6237045	B1	20010522	US 95399709	A	19950303	200130
			US 97929087	A	19970915	
CA 2143914	C	20020813	CA 2143914	A	19950303	200262

Priority Applications (No Type Date): JP 9541572 A 19950301; JP 9434701 A
19940304

Cited Patents: No-SR.Pub; 1.Jnl.Ref; EP 443768; EP 557205

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 670544	A2	E	41	G06F-009/44	
Designated States (Regional): DE GB					
CA 2143914	A			G06F-013/38	
JP 7295929	A		1	G06F-015/00	
EP 670544	A3			G06F-009/44	
US 6237045	B1			G06F-009/00	Cont of application US 95399709
CA 2143914	C	E		G06F-013/38	

Abstract (Basic): EP 670544 A

The information processing system **includes** a communication manager (10) interposed between a common platform (4) having a number of windows and an extensible Windows-based Elaboration Language WELL (11) system. The communication manager (10) **uses** a template to convert data between a visible **object** rendered in the common platform (4) and a data structure discernable by a computer. An extensible WELL kernel activates a required service module in response to an **instruction** entered by a user through the common platform and transmitted via the communication manager.

The common platform (4) **includes** a network window (5) which is used by the user (1) for controlling the entire system. An operation window (6) is used to render an **object** network having noun **objects** as nodes and verb **objects** as branches, and allows the user (1) to select any of the noun and verb **objects** . A command window (7) is used to render options and allows the client to select any of the options. A message window (8) is used to render messages sent to the system to the

client, and allows the client to enter responses if necessary, and a data window (9) is used to render the results of **transactions** and allows the client to **designate** points, segments or any of the entities.

USE /ADVANTAGE - Interaction between user and machine via common platform having multiple windows. Provides dynamic expansibility to any **different field** using WELL system.

Dwg.1/27

Title Terms: INTERACT; INFORMATION; PROCESS; SYSTEM; AGENT; CAPABLE;
GENERATE ; COMMON; PLATFORM; COMPOSE; MULTIPLE; WINDOW; CONVERT; DATA;
VISIBLE; **OBJECT** ; RENDER; COMMON; PLATFORM; DATA; STRUCTURE; DISCERNIBLE
; COMPUTER

Derwent Class: T01

International Patent Class (Main): **G06F-009/00** ; **G06F-009/44** ;
G06F-013/38 ; **G06F-015/00**

International Patent Class (Additional): **G06F-015/16** ; **G06F-017/00** ;
G06F-017/30

File Segment: EPI

29/5/29 (Item 27 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

010026811 **Image available**

WPI Acc No: 1994-294524/199436

XRPX Acc No: N94-231643

Framework processor of object -oriented application - uses system of dynamically linking load modules at run time, containing functions, static data, and classes

Patent Assignee: TALIGENT INC (TALI-N); OBJECT TECHNOLOGY LICENSING CORP (OBJE-N)

Inventor: GOLDSMITH D B; HENINGER A G; MOELLER C P; SCHAEFFER A; HENNINGER A G

Number of Countries: 046 Number of Patents: 008

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9419740	A1	19940901	WO 94US259	A	19940106	199436 B
AU 9462285	A	19940914	AU 9462285	A	19940106	199502
EP 664902	A1	19950802	EP 94909441	A	19940106	199535
			WO 94US259	A	19940106	
EP 664902	B1	19960522	EP 94909441	A	19940106	199625
			WO 94US259	A	19940106	
US 5519862	A	19960521	US 9323767	A	19930226	199626
DE 69400204	E	19960627	DE 600204	A	19940106	199631
			EP 94909441	A	19940106	
			WO 94US259	A	19940106	
JP 8508355	W	19960903	JP 94518951	A	19940106	199704
			WO 94US259	A	19940106	
CA 2135518	C	19990323	CA 2135518	A	19940106	199930

Priority Applications (No Type Date): US 9323767 A 19930226

Cited Patents: 02Jnl.Ref

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 9419740 A1 E 76 G06F-009/44

Designated States (National): AT AU BB BG BR BY CA CH CZ DE DK ES FI GB
HU JP KP KR KZ LK LU LV MG MN MW NL NO NZ PL PT RO RU SD SE SK UA UZ VN

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL
OA PT SE

AU 9462285 A G06F-009/44 Based on patent WO 9419740

EP 664902 A1 E 76 G06F-009/44 Based on patent WO 9419740

Designated States (Regional): DE FR GB IT

EP 664902 B1 E 55 G06F-009/44 Based on patent WO 9419740

Designated States (Regional): DE FR GB IT

US 5519862 A 44 G06F-009/40

DE 69400204 E G06F-009/44 Based on patent EP 664902

JP 8508355 W 109 G06F-015/00 Based on patent WO 9419740
CA 2135518 C G06F-009/44 Based on patent WO 9419740

Abstract (Basic): WO 9419740 A

The appts. load multiple **object** -oriented applications for concurrent execution. It **uses** a **collection** of load modules, each of which **contains** functions, static data, and classes, to **make** the modules appear to be statically **linked** .

Code executing in one load module can perform operations on **code** in another load module, such as calling a function, or obtaining a pointer to a function; accessing or obtaining a pointer to static data; calling any public or protected member function of a class to obtain a pointer or access a public or protected data member of the class; or cast to any base of an **object** of a class.

ADVANTAGE - **Implements** functions, static data, and classes in more flexible manner than prior operating systems, by utilisation of more consistent interaction interface with user.

Dwg.1A/27

Title Terms: FRAMEWORK; PROCESSOR; **OBJECT** ; ORIENT; **APPLY** ; SYSTEM; DYNAMIC; **LINK** ; LOAD; MODULE; RUN; TIME; **CONTAIN** ; FUNCTION; STATIC; DATA; CLASS

Derwent Class: T01

International Patent Class (Main): **G06F-009/40** ; **G06F-009/44** ; **G06F-015/00**

International Patent Class (Additional): **G06F-009/445** ; **G06F-009/46**

File Segment: EPI

29/5/30 (Item 28 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

010012253 **Image available**

WPI Acc No: 1994-279965/199434

Related WPI Acc No: 1996-465193; 1996-477304; 1999-132511

XRPX Acc No: N94-220527

Model information control system - has information models comprising dictionary of attributes, and objects whose attributes include control expressions that define process flow

Patent Assignee: TALATI K K (TALA-I)

Inventor: TALATI K K

Number of Countries: 022 Number of Patents: 010

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9418629	A1	19940818	WO 94US1562	A	19940210	199434 B
AU 9463920	A	19940829	AU 9463920	A	19940210	199501
			WO 94US1562	A	19940210	
US 5390330	A	19950214	US 9316430	A	19930211	199512
EP 686285	A1	19951213	EP 94911383	A	19940210	199603
			WO 94US1562	A	19940210	
EP 686285	A4	19960117	EP 94911383	A		199633
AU 677835	B	19970508	AU 9463920	A	19940210	199727
EP 686285	B1	20001220	EP 94911383	A	19940210	200105
			WO 94US1562	A	19940210	
DE 69426446	E	20010125	DE 626446	A	19940210	200112
			EP 94911383	A	19940210	
			WO 94US1562	A	19940210	
ES 2152308	T3	20010201	EP 94911383	A	19940210	200112
CA 2155865	C	20011002	CA 2155865	A	19940210	200161
			WO 94US1562	A	19940210	

Priority Applications (No Type Date): US 9316430 A 19930211

Cited Patents: US 4656603; US 5212650; US 5235701; US 5267175; EP 445769; WO 9102307

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 9418629 A1 E 49 G06F-015/00
 Designated States (National): AU CA CN JP KR
 Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL
 PT SE

AU 9463920 A G06F-015/00 Based on patent WO 9418629
 US 5390330 A 21 G06F-015/00
 EP 686285 A1 E 49 G06F-015/00 Based on patent WO 9418629
 Designated States (Regional): DE ES FR GB IT
 EP 686285 A4 G06F-015/00
 AU 677835 B G06F-015/00 Previous Publ. patent AU 9463920
 Based on patent WO 9418629

EP 686285 B1 E G06F-009/44 Based on patent WO 9418629
 Designated States (Regional): DE ES FR GB IT
 DE 69426446 E G06F-009/44 Based on patent EP 686285
 Based on patent WO 9418629

ES 2152308 T3 G06F-009/44 Based on patent EP 686285
 CA 2155865 C E G06F-017/30 Based on patent WO 9418629

Abstract (Basic): WO 9418629 A

The system (10) is used in conjunction with a user defined information model and one or more conventional information system program modules or functions to execute business applications. The model information control system **includes** an event-action-state machine (14) that manipulates the user defined information model (12) and the functions.

ADVANTAGE - No programming required since application developer need only write information model to **implement** application. Conventional software maintenance involving debugging, source **code** rewriting and testing is obviated when modification is to be made to application.

Dwg.1/2

Title Terms: MODEL; INFORMATION; CONTROL; SYSTEM; INFORMATION; MODEL; COMPRISE; DICTIONARY; ATTRIBUTE; **OBJECT** ; ATTRIBUTE; CONTROL; EXPRESS; DEFINE; PROCESS; FLOW

Derwent Class: T01

International Patent Class (Main): **G06F-009/44 ; G06F-015/00 ; G06F-017/30**

International Patent Class (Additional): **G06F-015/20 ; G06F-015/40 ; G06F-015/60**

File Segment: EPI

29/5/31 (Item 29 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

009172800 **Image available**

WPI Acc No: 1992-300234/199236

XRPX Acc No: N92-229901

Hierarchical distributed knowledge based machine initiated maintenance system - identifies failure in customer equipment, isolates error source using rules , hypotheses and collected data to isolate failed field replaceable unit causing error

Patent Assignee: STORAGE TECHNOLOGY CORP (STOS)

Inventor: HILL F L; HOLM D A; JURESTOVSKY N R; PINKHAM G A

Number of Countries: 019 Number of Patents: 008

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9214207	A1	19920820	WO 92US980	A	19920204	199236 B
AU 9215640	A	19920907	AU 9215640	A	19920204	199249
			WO 92US980	A	19920204	
EP 570513	A1	19931124	EP 92906769	A	19920204	199347
			WO 92US980	A	19920204	
US 5404503	A	19950404	US 91650931	A	19910205	199519
			US 9341146	A	19930330	
AU 661685	B	19950803	AU 9215640	A	19920204	199539
EP 570513	A4	19960626	EP 92906769	A	19920000	199644

EP 570513	B1	19990421	EP 92906769	A	19920204	199920
			WO 92US980	A	19920204	
DE 69228986	E	19990527	DE 628986	A	19920204	199927
			EP 92906769	A	19920204	
			WO 92US980	A	19920204	

Priority Applications (No Type Date): US 91650931 A 19910205; US 9341146 A 19930330

Cited Patents: US 3938830; US 4554661; US 4972367; US 4972453; US 5090014; 1.Jnl.Ref; AU 3088889; WO 9007152

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9214207	A1	E	51	G06F-011/00	
					Designated States (National): AU CA JP
					Designated States (Regional): AT BE CH DE DK ES FR GB GR IT LU MC NL SE
AU 9215640	A			G06F-011/00	Based on patent WO 9214207
EP 570513	A1	E	2	G06F-011/00	Based on patent WO 9214207
					Designated States (Regional): AT BE CH DE DK ES FR GB GR IT LI LU MC NL SE
US 5404503	A		18	G06F-011/00	Cont of application US 91650931
AU 661685	B			G06F-011/00	Previous Publ. patent AU 9215640
					Based on patent WO 9214207
EP 570513	B1	E		G06F-011/00	Based on patent WO 9214207
					Designated States (Regional): AT BE CH DE DK ES FR GB GR IT LI LU MC NL SE
DE 69228986	E			G06F-011/00	Based on patent EP 570513
					Based on patent WO 9214207
EP 570513	A4			G06F-011/00	

Abstract (Basic): WO 9214207 A

Maintenance systems (1, 10, 30) **configured** in a **tiered** hierarchy are **connected** to customer systems to detect failure in the customer system using detection circuits (20) and **collect** data using an **internal** expert system (11). A relational **object** database (12) maintains a failure analysis history **containing** data identifying the **field** replaceable unit and **associated** failure data.

The stored data is compared with failure analysis history and a likely **field** replaceable unit selected. If a unit cannot be identified a communication **connection** (51) is established with an **associated** hierarchically senior maintenance system.

USE /ADVANTAGE - Hierarchical distributed knowledge based system used to **implement** machine initiated maintenance capability. Provides efficient and timely maintenance of customer equipment.

Dwg.1/8

Title Terms: HIERARCHY; DISTRIBUTE; BASED; MACHINE; INITIATE; MAINTAIN; SYSTEM; IDENTIFY; FAIL; CUSTOMER; EQUIPMENT; ISOLATE; ERROR; SOURCE; **RULE ; COLLECT ; DATA; ISOLATE; FAIL; FIELD ; REPLACE; UNIT; CAUSE; ERROR**

Derwent Class: T01

International Patent Class (Main): **G06F-011/00**

International Patent Class (Additional): **G06F-009/44 ; G06F-011/22**

File Segment: EPI

29/5/32 (Item 30 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

008598202 **Image available**

WPI Acc No: 1991-102234/199114

XRPX Acc No: N91-078995

High level computer interface for program development - pushes editing and validation, error processing, looping, selection, ordering and auditing down into data access

Patent Assignee: AMDAHL CORP (AMDA)

Inventor: CHONG D T; KNUDSEN H; PLAZAK Z; ROBERTSON M; TAUGHER J E; YAFFE J ; TAUGHER J

Number of Countries: 017 Number of Patents: 043

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9103791	A	19910321				199114	B
AU 9064293	A	19910408				199127	
EP 489861	A1	19920617	EP 90914340	A	19900904	199225	
			WO 90US5007	A	19900904		
JP 5502527	W	19930428	JP 90513420	A	19900904	199322	
			WO 90US5007	A	19900904		
AU 9346083	A	19931216	AU 9064293	A	19900904	199406	
			AU 9346083	A	19930903		
AU 9346084	A	19931216	AU 9064293	A	19900904	199406	
			AU 9346084	A	19930903		
AU 9346082	A	19931223	AU 9064293	A	19900904	199407	
			AU 9346082	A	19930903		
EP 588445	A2	19940323	EP 90914340	A	19900904	199412	
			EP 93203195	A	19900904		
EP 588446	A2	19940323	EP 90914340	A	19900904	199412	
			EP 93203242	A	19900904		
EP 588447	A2	19940323	EP 90914340	A	19900904	199412	
			EP 93203243	A	19900904		
AU 646408	B	19940224	AU 9064293	A	19900904	199413	
EP 588445	A3	19940504	EP 93203195	A	19900904	199523	
EP 489861	A4	19930804	EP 90914340	A		199527	
EP 588446	A3	19951115	EP 93203242	A	19900904	199618	
EP 588447	A3	19960417	EP 90914340	A	19900904	199626	
			EP 93203243	A	19900904		
AU 671137	B	19960815	AU 9064293	A	19900904	199641	
			AU 9346082	A	19930903		
AU 671138	B	19960815	AU 9064293	A	19900904	199641	
			AU 9346084	A	19930903		
AU 673682	B	19961121	AU 9064293	A	19900904	199703	
			AU 9346083	A	19930903		
US 5584026	A	19961210	US 89402862	A	19890901	199704	
			US 89450298	A	19891213		
			US 92830548	A	19920131		
			US 92968237	A	19921029		
			US 9329699	A	19930311		
			US 95426489	A	19950420		
US 5586329	A	19961217	US 89402862	A	19890901	199705	
			US 89450298	A	19891213		
			US 92830548	A	19920131		
			US 92968237	A	19921029		
			US 9329908	A	19930311		
			US 95424234	A	19950418		
US 5586330	A	19961217	US 89402862	A	19890901	199705	
			US 89450298	A	19891213		
			US 92830548	A	19920131		
			US 92968237	A	19921029		
			US 9329478	A	19930311		
			US 95424241	A	19950418		
US 5594899	A	19970114	US 89402862	A	19890901	199709	
			US 89450298	A	19891213		
			US 92830548	A	19920131		
			US 92968237	A	19921029		
			US 9329902	A	19930311		
			US 94347588	A	19941201		
US 5596752	A	19970121	US 89402862	A	19890901	199710	
			US 89450298	A	19891213		
			US 92830548	A	19920131		
			US 92968237	A	19921029		
			US 9329700	A	19930311		
EP 489861	B1	19970709	EP 90914340	A	19900904	199732	
			WO 90US5007	A	19900904		
DE 69031040	E	19970814	DE 631040	A	19900904	199738	
			EP 90914340	A	19900904		
			WO 90US5007	A	19900904		

US 5682535	A	19971028	US 89402862	A	19890901	199749
			US 92830550	A	19920131	
			US 92968474	A	19921029	
			US 9397096	A	19930726	
			US 97784736	A	19970113	
EP 588445	B1	19990519	EP 90914340	A	19900904	199924
			EP 93203195	A	19900904	
EP 588447	B1	19990519	EP 90914340	A	19900904	199924
			EP 93203243	A	19900904	
DE 69033120	E	19990624	DE 633120	A	19900904	199931
			EP 93203195	A	19900904	
DE 69033121	E	19990624	DE 633121	A	19900904	199931
			EP 93203243	A	19900904	
EP 588446	B1	19990707	EP 90914340	A	19900904	199931
			EP 93203242	A	19900904	
DE 69033203	E	19990812	DE 633203	A	19900904	199938
			EP 93203242	A	19900904	
ES 2132175	T3	19990816	EP 93203195	A	19900904	199939
ES 2132176	T3	19990816	EP 93203243	A	19900904	199939
ES 2133145	T3	19990901	EP 93203242	A	19900904	199941
CA 2284245	A1	19910302	CA 2066724	A	19900904	200015
			CA 2284245	A	19900904	
CA 2284247	A1	19910302	CA 2066724	A	19900904	200015
			CA 2284247	A	19900904	
CA 2284248	A1	19910302	CA 2066724	A	19900904	200015
			CA 2284248	A	19900904	
CA 2284250	A1	19910302	CA 2066724	A	19900904	200015
			CA 2284250	A	19900904	
CA 2066724	C	20001205	CA 2066724	A	19900904	200101
			WO 90US5007	A	19900904	
CA 2284245	C	20010206	CA 2066724	A	19900904	200111
			CA 2284245	A	19900904	
CA 2284250	C	20011204	CA 2066724	A	19900904	200203
			CA 2284250	A	19900904	
CA 2284248	C	20011204	CA 2066724	A	19900904	200203
			CA 2284248	A	19900904	

Priority Applications (No Type Date): US 89450298 A 19891213; US 89402862 A 19890901; US 92830548 A 19920131; US 92968237 A 19921029; US 9329699 A 19930311; US 95426489 A 19950420; US 9329908 A 19930311; US 95424234 A 19950418; US 9329478 A 19930311; US 95424241 A 19950418; US 9329902 A 19930311; US 94347588 A 19941201; US 9329700 A 19930311; US 92830550 A 19920131; US 92968474 A 19921029; US 9397096 A 19930726; US 97784736 A 19970113

Cited Patents: 1.Jnl.Ref; EP 163577; US 4791561; EP 331060; US 4860204; DE 3503119; EP 243110; GB 2126761; US 4099230

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

WO 9103791	A				
Designated States (National): AU CA JP US					
Designated States (Regional): AT BE CH DE DK ES FR GB IT LU NL SE					
EP 489861	A1	E	2	G06F-015/40	Based on patent WO 9103791
Designated States (Regional): AT BE CH DE DK ES FR GB IT LI LU NL SE					
JP 5502527	W			G06F-009/06	Based on patent WO 9103791
AU 9346083	A			G06F-009/44	Div ex application AU 9064293
AU 9346084	A			G06F-015/40	Div ex application AU 9064293
AU 9346082	A			G06F-009/45	Div ex application AU 9064293
EP 588445	A2	E	84	G06F-015/413	Related to application EP 90914340
Designated States (Regional): AT BE CH DE DK ES FR GB IT LI LU NL SE					
EP 588446	A2	E	84	G06F-009/44	Related to application EP 90914340
Designated States (Regional): AT BE CH DE DK ES FR GB IT LI LU NL SE					
EP 588447	A2	E	87	G06F-009/44	Related to application EP 90914340
Designated States (Regional): AT BE CH DE DK ES FR GB IT LI LU NL SE					
AU 646408	B			G06F-015/40	Previous Publ. patent AU 9064293
Based on patent WO 9103791					
EP 588446	A3				Related to patent EP 489861
EP 588447	A3				Div ex application EP 90914340

AU 671137	B	G06F-009/45	Div ex application AU 9064293 Previous Publ. patent AU 9346082
AU 671138	B	G06F-015/40	Div ex application AU 9064293 Previous Publ. patent AU 9346084
AU 673682	B	G06F-009/44	Div ex application AU 9064293 Previous Publ. patent AU 9346083
US 5584026	A	72 G06F-017/30	CIP of application US 89402862 Cont of application US 89450298 Cont of application US 92830548 Div ex application US 92968237 Cont of application US 9329699
US 5586329	A	268 G06F-009/45	CIP of application US 89402862 Cont of application US 89450298 Cont of application US 92830548 Div ex application US 92968237 Cont of application US 9329908
US 5586330	A	74 G06F-009/45	CIP of application US 89402862 Cont of application US 89450298 Cont of application US 92830548 Div ex application US 92968237 Cont of application US 9329478
US 5594899	A	72 G06F-017/30	CIP of application US 89402862 Cont of application US 89450298 Cont of application US 92830548 Div ex application US 92968237 Cont of application US 9329902
US 5596752	A	73 G06F-015/00	CIP of application US 89402862 Cont of application US 89450298 Cont of application US 92830548 Div ex application US 92968237
EP 489861	B1 E	89 G06F-009/44	Based on patent WO 9103791
Designated States (Regional): AT			BE CH DE DK ES FR GB IT LI LU NL SE
DE 69031040	E	G06F-009/44	Based on patent EP 489861 Based on patent WO 9103791
US 5682535	A	75 G06F-009/44	Cont of application US 89402862 Cont of application US 92830550 Cont of application US 92968474 Cont of application US 9397096
EP 588445	B1 E	G06F-009/44	Div ex application EP 90914340 Div ex patent EP 489861
Designated States (Regional): AT			BE CH DE DK ES FR GB IT LI LU NL SE
EP 588447	B1 E	G06F-009/44	Div ex application EP 90914340 Div ex patent EP 489861
Designated States (Regional): AT			BE CH DE DK ES FR GB IT LI LU NL SE
DE 69033120	E	G06F-009/44	Based on patent EP 588445
DE 69033121	E	G06F-009/44	Based on patent EP 588447
EP 588446	B1 E	G06F-009/44	Div ex application EP 90914340 Div ex patent EP 489861
Designated States (Regional): AT			BE CH DE DK ES FR GB IT LI LU NL SE
DE 69033203	E	G06F-009/44	Based on patent EP 588446
ES 2132175	T3	G06F-009/44	Based on patent EP 588445
ES 2132176	T3	G06F-009/44	Based on patent EP 588447
ES 2133145	T3	G06F-009/44	Based on patent EP 588446
CA 2284245	A1 E	G06F-017/30	Div ex application CA 2066724
CA 2284247	A1 E	G06F-009/45	Div ex application CA 2066724
CA 2284248	A1 E	G06F-009/45	Div ex application CA 2066724
CA 2284250	A1 E	G06F-009/45	Div ex application CA 2066724
CA 2066724	C E	G06F-015/40	Based on patent WO 9103791
CA 2284245	C E	G06F-017/30	Div ex application CA 2066724
CA 2284250	C E	G06F-009/45	Div ex application CA 2066724
CA 2284248	C E	G06F-009/45	Div ex application CA 2066724

Abstract (Basic): WO 9103791 A

Objects including **rules** and data are stored in buffers identified by a buffer address. A current **rule** including a static data area and a modifiable data area is executed. The static data area stores **object** identifiers with offsets to positions in the modifiable

data area at which buffer addresses of buffers storing identified
objects are to be located at execution time.

Objects are found in the buffers in response to the executing
device, and buffer addresses are supplied to the modifiable data area
at execution time.

ADVANTAGE - Frees programmer of explicit recognition in program of
environmental parameters. (2pp Dwg.No.1/21

Title Terms: HIGH; LEVEL; COMPUTER; INTERFACE; PROGRAM; DEVELOP; PUSH; EDIT
; VALID; ERROR; PROCESS; LOOP; SELECT; ORDER; AUDIT; DOWN; DATA; ACCESS

Derwent Class: T01

International Patent Class (Main): G06F-009/06 ; G06F-009/44 ;

G06F-009/45 ; G06F-015/00 ; G06F-015/40 ; G06F-015/413 ; G06F-017/30

International Patent Class (Additional): G06F-015/20

File Segment: EPI

29/5/33 (Item 31 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

007849421 **Image available**

WPI Acc No: 1989-114533/198915

XRPX Acc No: N89-087472

**Expert system with process control - has collection of interference
rules presented in constrained formats for user comprehension**

Patent Assignee: DU PONT DE NEMOURS & CO E I (DUPO)

Inventor: SKEIRIK R D; DECARIA F O; DE CARIA F O; CARIA F O; SKEIRIK R

Number of Countries: 017 Number of Patents: 019

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 8903092	A	19890406	WO 88US3360	A	19880930	198915 B
AU 8826067	A	19890418				198930
EP 335957	A	19891011	EP 88909446	A	19880930	198941
BR 8807224	A	19891031				198949
US 4884217	A	19891128	US 87102832	A	19870930	199006
US 4907167	A	19900306	US 87103118	A	19870930	199016
US 4910691	A	19900320	US 87103124	A	19870930	199017
US 4920499	A	19900424	US 87103050	A	19870930	199021
JP 2501420	W	19900517	JP 88508686	A	19880930	199026
US 4965742	A	19901023	US 87103047	A	19870930	199045
US 5006992	A	19910409	US 87103014	A	19870930	199117
CA 1297558	C	19920317				199217
CA 1297559	C	19920317				199217
CA 1297560	C	19920317				199217
CA 1297561	C	19920317				199217
CA 1309182	C	19921020	CA 578704	A	19880928	199248
CA 1310424	C	19921117	CA 578705	A	19880928	199252
EP 335957	B1	19991117	EP 88909446	A	19880930	199953
			WO 88US3360	A	19880930	
DE 3856379	G	19991223	DE 3856379	A	19880930	200006
			EP 88909446	A	19880930	
			WO 88US3360	A	19880930	

Priority Applications (No Type Date): US 87103124 A 19870930; US 87102832 A
19870930; US 87103014 A 19870930; US 87103047 A 19870930; US 87103050 A
19870930; US 87103118 A 19870930

Cited Patents: 04 64278200; 04 64804400; 04 65837000; 04 67084800; 4713775

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 8903092 A E 305

Designated States (National): AU BR JP KR

Designated States (Regional): AT BE CH DE FR GB IT LU NL SE

DE 3856379 G G06F-009/44 Based on patent EP 335957

Based on patent WO 8903092

EP 335957 A E

Designated States (Regional): AT BE CH DE FR GB IT LI LU NL SE

US 4884217 A 91

US 4907167 A 75
 US 4910691 A 120
 US 4920499 A 89
 US 4965742 A 92
 EP 335957 B1 E G06F-009/44 Based on patent WO 8903092
 Designated States (Regional): AT BE CH DE FR GB IT LI LU NL SE
 CA 1309182 C G06F-015/46
 CA 1310424 C G06F-015/46

Abstract (Basic): WO 8903092 A

The **rules** are of three classes: 1) retrieval **rules** (210), which each **associate** one of several **attributes** to an **object** in accordance with the values of inputs; 2) analysis **rules** (220), which selectively **associate** an **attribute** with an **object**, and which are somewhat analogous to the natural-language inference **rules** which would be used in communications between domain experts; and 3) action **rules** (230), which selectively carry out the output and control actuation options, based on the **attributes associated with objects** by the **other rules**. The expert system can be used in conjunction with an integrated system for process control (150), which itself can exist as a separate, discrete entity.

The system **contains** a process supervisor procedure (130) (which is preferably the top-level procedure) which can define parameters for one or more controller systems (or control procedures). The supervisor procedure can also be **configured** as a modular software structure with modules which can be revised by a user at any time, without significantly interrupting the operation of the process supervisor. Users can define or redefine modules by editing highly constrained templates

Title Terms: EXPERT; SYSTEM; PROCESS; CONTROL; **COLLECT**; INTERFERENCE;
RULE; PRESENT; CONSTRAIN; FORMAT; USER; COMPREHENSIVE

Derwent Class: T01; T06

International Patent Class (Main): **G06F-009/44**; **G06F-015/46**

International Patent Class (Additional): G05B-011/32; G05B-013/02;

G05B-019/41

File Segment: EPI

29/5/34 (Item 32 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

004186766

WPI Acc No: 1985-013646/198503

XRPX Acc No: N85-009708

Digital computer with single machine cycle instruction set - performs full shift, merge, insert and bit alignment functions

Patent Assignee: IBM CORP (IBMC)

Inventor: HAO H T; MARKSTEIN P W; RADIN G

Number of Countries: 004 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 130380	A	19850109	EP 84106177	A	19840530	198503 B
US 4569016	A	19860204	US 83509836	A	19830630	198608

Priority Applications (No Type Date): US 83509836 A 19830630

Cited Patents: A3...8745; No-SR.Pub; US 3906459; US 3982229; US 4180861; US 4467444

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 130380	A	E	83		

Designated States (Regional): DE FR GB

Abstract (Basic): EP 130380 A

The system **includes** a main memory, a high speed cache and a CPU, the latter including an arithmetic and logic unit, and construction unit and a condition **code generator**. A mechanism is provided for performing single machine cycle executable Mask and Rotate

instructions in the CPU, and **includes** a unit (62) operable in parallel for **generating** an N bit mask word from transition points address specified by an **instruction** .

The mechanism also **includes** a unit (60) for rotating the N-bit subject operand which is to be partially merged with an N-bit subject operand by an amount specified in the **instruction** . Another unit (66) merges the rotated subject operand into the **object** operand under control of the mask so that data representative of the **object** operand is retained where the mask has a first binary content and data representative of the rotated subject operand is retained when the mask has the opposite binary content. A unit (R-2) returns the retained merged N-bit word to a register in the CPU specified by the **instructions** .

ADVANTAGE - Any **designated field** in one operand may be inserted in and replace any **designated field** of a **second** operand.

3/3

Title Terms: DIGITAL; COMPUTER; SINGLE; MACHINE; CYCLE; **INSTRUCTION** ; SET; PERFORMANCE; FULL; SHIFT; MERGE; INSERT; BIT; ALIGN; FUNCTION

Derwent Class: T01

International Patent Class (Additional): G06F-009/30

File Segment: EPI

Set	Items	Description
S1	742505	OBJECT? OR CUSTOM(1W)CONTROL? OR SOFTWARE()ROUTINE? OR DLL OR DLLS OR CODE()MODULE? OR VBX OR VBXS OR VISUAL()BASIC()EXTENSION? OR READY()MADE() (COMPONENT? OR CONTROL?) OR OCX OR OCXS OR (ACTIVEX OR ACTIVE()X)()CONTROL?
S2	1390233	GENERAT? OR MAKE? ? OR MAKING OR CREATE? OR CREATING OR PRODUC?
S3	353185	SCRIPT? OR INSTRUCTION? OR RULE? OR SYNTAX OR CODE OR CODES
S4	8411	(MAINTAIN? OR PRESERVE? OR KEEP? ? OR SUSTAIN? OR SUPPORT?-(3N) (SEPARATELY OR INDEPENDENTLY OR INDIVIDUALLY OR SINGLY)
S5	1379361	WRAP? OR CONTAIN? OR INCLUDE? OR HOLD? OR ENCLOSE?
S6	1112278	PROPERTY OR PROPERTIES OR FIELD? OR PATTERN? OR QUALITY OR QUALITIES OR ATTRIBUTE? OR FEATURE?
S7	708577	EVENT? OR HAPPENING? OR INCIDENT? OR OCCURRENCE? OR DEVELOPMENT? OR EPISODE?
S8	77292	(ADDITIONAL OR ANOTHER OR SECOND OR DIFFERENT) (10N) (S7 OR -S7)
S9	1117860	ABOVE OR BEYOND OR EXCEED? OR SURPASS? OR OVER()REACH? OR -OVERREACH?
S10	1469654	INTERNAL OR CONFIGUR? OR DESIGN? OR ARRANGE? OR SET()UP OR SETUP OR PROGRAM?()IN OR SETTING? OR IMBED? OR EMBED? OR FIXED OR ENTRENCH? OR FASTEN? OR INFIX? OR INGRAIN
S11	1285855	UTILIZE OR APPLY? OR APPLIES OR EMPLOY? OR IMPLEMENT? OR USE OR USES?
S12	1457942	LINK? OR (PUT OR LAY OR PIECE)()TOGETHER OR JOIN? OR UNITE? OR UNIFY OR UNIFIES OR CONNECT? OR ASSEMBLE? OR COLLECT? OR -COMBINE? OR ASSOCIAT? OR AFFILIAT? OR TIE? OR TRANSACTION?
S13	365119	SEQUENCE? OR QUEUE OR QUEUED OR QUEUING OR QUEUEING OR RANK? OR (LOGICAL OR SERIAL)()ORDER OR CONSECUTIVE?
S14	18596	S1 (S) S2 (S) S3
S15	337345	(ADDITIONAL OR ANOTHER OR SECOND OR DIFFERENT) (10N) (S6 OR -S7)
S16	119728	S5 (S) S15
S17	53237	(S6 OR S7) (S) S10 (S) S1
S18	104523	S11 (S) S15
S19	214244	(S12 OR S13) (S) S1
S20	24	S14 (S) S4
S21	476	S14 (S) S16 (S) S17 (S) S18 (S) S19
S22	350	S14 (10N) S16 (10N) S17 (10N) S18 (10N) S19
S23	369	S20 OR S22
S24	199	S23 AND IC=G06F?
S25	52	S24 AND IC=G06F-009?

File 348:EUROPEAN PATENTS 1978-2003/Nov W05

(c) 2003 European Patent Office

File 349:PCT FULLTEXT 1979-2002/UB=20031211,UT=20031204

(c) 2003 WIPO/Univentio

25/5,K/11 (Item 11 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00663613

METHOD AND APPARATUS FOR ENTERPRISE DESKTOP MANAGEMENT
VERFAHREN UND VORRICHTUNG ZUM VERWALTEN VON TISCHCOMPUTERN EINES
UNTERNEHMENS
PROCEDE ET APPAREIL DESTINES A LA GESTION DU SYSTEME INFORMATIQUE DANS
L'ENTREPRISE

PATENT ASSIGNEE:

NOVADIGM, INC., (1871910), Suite 200, One International Boulevard,,
Mahwah, NJ 07495, (US), (applicant designated states:
AT;BE;CH;DE;DK;ES;FR;GB;GR;IE;IT;LI;LU;MC;NL;PT;SE)

INVENTOR:

FITZGERALD, Albion, J., 200 Melrose Place, Ridgewood, NJ 07450, (US)
FITZGERALD, Joseph, J., 24 Bonticou View Drive, New Paltz, NY 12561, (US)

LEGAL REPRESENTATIVE:

Wright, Hugh Ronald (38051), Brookes & Martin 52/54 High Holborn, London
WC1V 6SE, (GB)

PATENT (CC, No, Kind, Date): EP 696368 A1 960214 (Basic)
EP 696368 B1 970730
WO 9425924 941110

APPLICATION (CC, No, Date): EP 94916651 940502; WO 94US4896 940502

PRIORITY (CC, No, Date): US 56333 930430

DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; LU; MC;
NL; PT; SE

INTERNATIONAL PATENT CLASS: G06F-017/60 ; G06F-009/44

NOTE:

No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse: 020626 B1 Date of lapse of European Patent in a
contracting state (Country, date): AT
19970730, DK 19970730, ES 19970730, GR
19970730, PT 19971107,

Lapse: 20000202 B1 Date of lapse of European Patent in a
contracting state (Country, date): AT
19970730, DK 19970730, GR 19970730, PT
19971107,

Application: 950201 A International application (Art. 158(1))

Application: 960214 A1 Published application (A1with Search Report
;A2without Search Report)

Examination: 960214 A1 Date of filing of request for examination:
951030

Examination: 960501 A1 Date of despatch of first examination report:
960313

Grant: 970730 B1 Granted patent

Lapse: 980408 B1 Date of lapse of the European patent in a
Contracting State: AT 970730, DK 970730

Lapse: 980408 B1 Date of lapse of the European patent in a
Contracting State: AT 970730, DK 970730

Lapse: 980610 B1 Date of lapse of the European patent in a
Contracting State: AT 970730, DK 970730, PT
971107

Oppn None: 980722 B1 No opposition filed

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9707W5	3039
CLAIMS B	(German)	9707W5	2454
CLAIMS B	(French)	9707W5	3501
SPEC B	(English)	9707W5	12999

Total word count - document A 0

Total word count - document B 21993

Total word count - documents A + B 21993

INTERNATIONAL PATENT CLASS: G06F-017/60 ...

... G06F-009/44

...SPECIFICATION desktop configurations to largely generic groups of desktops.

An advantage of adaptive SH list generation using dynamic **linkage** substitution is that it can reduce the need for administrator intervention when the desktop environment is changed...

...resolution of an SH lists which reflect the particular needs of individual desktop environments without resort to **rule** based programming. The novel **use** of dynamic linkages, therefore, advantageously permits desktop users to customize their desktops to fit their particular requirements without imposing an undue management burden upon the network administrators. The **use** of dynamic linkages in connection with interrogation software that assigns values to the variables used to resolve such dynamic linkages, affords the automatic (administrator-less) connection of sub-lists (or **objects**) appropriate to individual desktops. These sub-lists (or **objects**) **contain** the information required to update the desktop during a **configuration event**.

Another advantage of SH list generation using dynamic linkage substitution is that the SH lists can automatically adapt to the policies and technology configurations of individual desktops. As explained below, the Schema can **contain** both dynamic linkages related to policy considerations and dynamic linkages related to technology considerations. During desktop configuration, a SH list is resolved from the Schema by determining which policy linkages and which technology linkages **apply** for the desktop in its current configuration and with its current user authorization level. The technology configuration...

25/5,K/12 (Item 12 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00662803

Method and apparatus for producing a composite second image in the spatial context of a first image
Verfahren und Gerat zur Erzeugung eines zweiten gemischten Bildsignals im raumlichen Kontext eines ersten Bildsignals
Procede et dispositif pour la production d'une deuxieme image composite dans un contexte spatial d'une premiere image

PATENT ASSIGNEE:

XEROX CORPORATION, (219783), Xerox Square, Rochester, New York 14644,
(US), (Proprietor designated states: all)

INVENTOR:

Stone, Maureen C., 191 Pine Lane, Los Altos CA 94022, (US)
Fishkin, Kenneth, P., 924 Haven Avenue, Redwood City CA 94063, (US)
Bier, Eric A., 175 Sherland Avenue, Mountain View CA 94043, (US)
DeRose, Anthony, 3018 NE 97th Street, Seattle WA 98195, (US)

LEGAL REPRESENTATIVE:

Grunecker, Kinkeldey, Stockmair & Schwanhausser Anwaltssozietat (100721)
, Maximilianstrasse 58, 80538 Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 636971 A2 950201 (Basic)
EP 636971 A3 960103
EP 636971 B1 011017

APPLICATION (CC, No, Date): EP 94305294 940719;

PRIORITY (CC, No, Date): US 96193 930721

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS: G06F-009/44 ; G06F-003/033 ; G06T-011/00

CITED PATENTS (EP B): EP 538715 A; EP 544509 A; US 5222159 A

CITED REFERENCES (EP B):

COMPUTER GRAPHICS PROCEEDINGS, PROCEEDING OF SIGGRAPH 20TH ANNUAL
INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE
TECHNIQUES. THE EYE OF TECHNOLOGY, ANAHEIM, CA, USA, 1-6 AUG. 1993,
ISBN 0-89791-601-8, 1993, NEW YORK, NY, USA, ACM, USA, pages 73-80,

ABSTRACT EP 636971 A2

A method is disclosed for operating a processor-controlled machine to produce a composite view of an original, or first, image by combining the functions of multiple viewing operations and using the model data structure from which the first image was produced (called the "FIMDS".) A first viewing operation region (1VOR) (20) in a first viewing position in the display area (180) of the display device has displayed in the 1VOR (20) a second image produced according to a first viewing operation (VO1), associated with the 1VOR, that maps the FIMDS to image definition data defining the second image. Request signal data is received from a signal source to present a second viewing operation region (2VOR) (40) in the display area (180) coextensive with at least a portion of the 1VOR (20), forming a composite viewing operation region in the overlapping region. The 2VOR (40) has a second viewing operation (VO2) associated with it for mapping an input model data structure, typically but not necessarily the FIMDS, to image definition data defining a second image for display in the 2VOR (2VOR-SI), providing an alternate view of the FIMDS that is different from that provided by the VO1 associated with the 1VOR. In response to the request signal data, a composite viewing operation, composed from the functions of the VO1 and the VO2, maps the FIMDS to image definition data defining a composite image for display in the composite viewing operation region substantially at the same time as the first image is being displayed in the display area. This gives the perception to the machine user of applying composed viewing operations to information presented in the original image. (see image in original document)

ABSTRACT WORD COUNT: 307

NOTE:

Figure number on first page: 4

LEGAL STATUS (Type, Pub Date, Kind, Text):

Grant: 011017 B1 Granted patent
Change: 20000329 A2 Legal representative(s) changed 20000210
Oppn None: 021009 B1 No opposition filed: 20020718
Application: 950201 A2 Published application (Alwith Search Report
;A2without Search Report)
Change: 951227 A2 Obligatory supplementary classification
(change)
Search Report: 960103 A3 Separate publication of the European or
International search report
Change: 960306 A2 Rectifications of patent applications (change)
Examination: 960828 A2 Date of filing of request for examination:
960703
Change: 980930 A2 Representative (change)
Examination: 981209 A2 Date of despatch of first examination report:
981027

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF2	5939
CLAIMS B	(English)	200142	2266
CLAIMS B	(German)	200142	1956
CLAIMS B	(French)	200142	2768
SPEC A	(English)	EPABF2	9217
SPEC B	(English)	200142	9198
Total word count - document A			15159
Total word count - document B			16188
Total word count - documents A + B			31347

INTERNATIONAL PATENT CLASS: G06F-009/44 ...

... G06F-003/033

...CLAIMS images to the display, and for accessing the data stored in the

memory; the processor further being **connected** for receiving data from the storage medium access device;

the article comprising:

a data storage medium that...

...the processor when the article is used in the machine; the stored data comprising instruction data indicating **instructions** the processor can execute;

the processor, in executing the **instructions**, receiving first request signal data from the user input device indicating a first display request by the...

...the present image position of a first image segment of the first image including the first display **feature**;

the processor further, in executing the **instructions**, responding to the first request signal data indicating the first display request by

producing a second model data structure using the first **feature** data item from the first image model data structure; the second model data structure including an output **feature** data item **produced** using the first **feature** data item;

producing the image definition data defining a **second** image for display in the 1VOR (1VOR-SI) using the **second** model data structure including the output **feature** data item; the 1VOR-SI including an output display **feature** representing the output feature data item, and having size and shape dimensions substantially identical to size and 1VOR; and

providing the image definition data defining the 1VOR-SI to the output circuitry **connected** to the display device so that the display device presents the 1VOR-SI, including the output display...

25/5,K/13 (Item 13 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

(c) 2003 European Patent Office. All rts. reserv.

00614518

SYSTEM FOR CHANGING SOFTWARE DURING COMPUTER OPERATION

SYSTEM ZUM ANDERN VON SOFTWARE WAHREND DES RECHNERBETRIEBS.

SYSTEME PERMETTANT LE REMPLACEMENT D'UN LOGICIEL PENDANT LE FONCTIONNEMENT D'UN ORDINATEUR

PATENT ASSIGNEE:

TELEFONAKTIEBOLAGET LM ERICSSON, (213761), , 126 25 Stockholm, (SE),
(Proprietor designated states: all)

INVENTOR:

NILSSON, Kjell, Rickard, Katarina Bangatan 71, 2trp, S-116 39 Stockholm,
(SE)

MARKSTROM, Ulf, Kurt, Haakan, Vanadisvagen 32, 3trp, S-113 46 Stockholm,
(SE)

KLOFVER, Jan, Leif, Ingemar, Baagspannarvagen 5, S-125 30 Alvsjo, (SE)

LEGAL REPRESENTATIVE:

Wideberg, Olle Sven et al (39494), Ericsson Telecom AB, IPR Management
and Patent Department, 126 25 Stockholm, (SE)

PATENT (CC, No, Kind, Date): EP 648353 A1 950419 (Basic)

EP 648353 B1 990915

WO 9401819 940120

APPLICATION (CC, No, Date): EP 93915035 930511; WO 93SE417 930511

PRIORITY (CC, No, Date): US 907294 920701

DESIGNATED STATES: CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; NL; SE

INTERNATIONAL PATENT CLASS: G06F-009/44 ; G06F-009/445 ; G06F-009/46

CITED PATENTS (EP B): WO 91/08534 A; GB 2242293 A; US 5008814 A; US 5155837

A

CITED REFERENCES (EP B):

National Electronics Conference Proceedings of the National ..., Volume
38, 1984, KENNETH Y. NIENG, "Updating Software in a Real-Time
Distributed System", page 57 - page 60.

PATENT ABSTRACTS OF JAPAN, Vol. 13, No. 572, P-978; & JP,A,01 239 633

(NEC CORP.), 25 Sept 1989 (25.09.89).
PATENT ABSTRACTS OF JAPAN, Vol. 14, No. 104, P-1013; & JP,A,01 307 825
(NEC CORP.), 12 December 1989 (12.12.89).
PATENT ABSTRACTS OF JAPAN, Vol. 12, No. 350, P-760; & JP,A,63 106 047
(NEC CORP.), 11 May 1988 (11.05.88).;

NOTE:

No A-document published by EPO
LEGAL STATUS (Type, Pub Date, Kind, Text):
Oppn None: 000830 B1 No opposition filed: 20000616
Application: 940427 A International application (Art. 158(1))
Application: 950419 A1 Published application (A1with Search Report
;A2without Search Report)
Examination: 950419 A1 Date of filing of request for examination:
941123
Examination: 980429 A1 Date of despatch of first examination report:
980311
Change: 980729 A1 Representative (change)
Grant: 990915 B1 Granted patent
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS B (English) 9937 3913
CLAIMS B (German) 9937 3672
CLAIMS B (French) 9937 4137
SPEC B (English) 9937 10301
Total word count - document A 0
Total word count - document B 22023
Total word count - documents A + B 22023

INTERNATIONAL PATENT CLASS: G06F-009/44 ...

... G06F-009/445 ...

... G06F-009/46

...SPECIFICATION the steps required in order to create objects within the system as shown also in FIG. 6.

Objects are language constructs that contain both data and **code** or functions within a single package or unit. Because they are able to contain both data and **code**, they act as miniature, independent programs. They can be used, therefore, as building blocks in **creating** more complex programs without having to redevelop the **code** necessary for those functions. Because they can be **maintained** and modified **independently**, program maintenance and revision is simplified.

A class is a template that is used to define an...

25/5,K/14 (Item 14 from file: 348)
DIALOG(R) File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00614517

SYSTEM AND METHOD FOR DYNAMIC RUN-TIME BINDING OF SOFTWARE MODULES IN A COMPUTER SYSTEM

SYSTEM UND VERFAHREN ZUR DYNAMISCHEN LAUFZEIT-BINDUNG VON SOFTWARE-MODULEN IN EINEM RECHNERSYSTEM.

SYSTEME ET METHODE POUR LA LIAISON DYNAMIQUE D'EXECUTION DE MODULES LOGICIELS DANS UN SYSTEME INFORMATIQUE

PATENT ASSIGNEE:

TELEFONAKTIEBOLAGET LM ERICSSON, (213761), , 126 25 Stockholm, (SE),
(Proprietor designated states: all)

INVENTOR:

MARKSTROM, Ulf, Kurt, Haakan, Vanadisvagen 32, 3trp, S-113 46 Stockholm,
(SE)

LUNDIN, Lars, Kenneth, Skimmelvagen 4, S-152 57 Sodertalje, (US)

LEGAL REPRESENTATIVE:

Wideberg, Olle Sven et al (39494), Ericsson Telecom AB, IPR Management

and Patent Department, 126 25 Stockholm, (SE)
PATENT (CC, No, Kind, Date): EP 648352 A1 950419 (Basic)
EP 648352 B1 991208
WO 9401818 940120
APPLICATION (CC, No, Date): EP 93915034 930511; WO 93SE416 930511
PRIORITY (CC, No, Date): US 907307 920701
DESIGNATED STATES: CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; NL; SE
INTERNATIONAL PATENT CLASS: G06F-009/44
CITED PATENTS (EP B): EP 518195 A; GB 2242293 A; GB 2258068 A; US 5093916 A
; US 5175828 A

CITED REFERENCES (EP B):

PATENT ABSTRACTS OF JAPAN, Vol. 14, No. 104, P-1013; & JP,A,01 307 825
(NEC CORP), 12 December 1989 (12.12.89).

PATENT ABSTRACTS OF JAPAN, Vol. 13, No. 572, P-978; & JP,A,01 239 633
(NEC CORP), 25 Sept 1989 (25.09.89).;

NOTE:

No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):

Oppn None: 001122 B1 No opposition filed: 20000909
Application: 940427 A International application (Art. 158(1))
Application: 950419 A1 Published application (A1with Search Report
;A2without Search Report)
Examination: 950419 A1 Date of filing of request for examination:
941123
Examination: 980506 A1 Date of despatch of first examination report:
980318
Change: 980729 A1 Representative (change)
Change: 990324 A1 Title of invention (German) (change)
Change: 990324 A1 Title of invention (English) (change)
Change: 990324 A1 Title of invention (French) (change)
Change: 990331 A1 Title of invention (German) (change)
Change: 990331 A1 Title of invention (English) (change)
Change: 990331 A1 Title of invention (French) (change)
Change: 991208 A1 Title of invention (German) changed: 19991015
Grant: 991208 B1 Granted patent

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9949	2198
CLAIMS B	(German)	9949	2090
CLAIMS B	(French)	9949	2648
SPEC B	(English)	9949	8272
Total word count - document A			0
Total word count - document B			15208
Total word count - documents A + B			15208

INTERNATIONAL PATENT CLASS: G06F-009/44

...SPECIFICATION the steps required in order to create objects within the system as shown also in FIG. 4.

Objects are run-time instances of classes that contain definitions of both data and functions within a single package or unit. Because they are able to contain both data and **code**, they act as miniature, independent programs. They can be used, therefore, as building blocks in **creating** more complex programs without having to redevelop the **code** necessary for those functions. Because they can be **maintained** and modified **independently**, program maintenance and revision is simplified.

A class is a template that is used to define an...

25/5,K/15 (Item 15 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00555558

APPARATUS AND METHOD FOR CREATION OF A USER DEFINABLE VIDEO DISPLAYED
DOCUMENT SHOWING CHANGES IN REAL TIME DATA

VORRICHTUNG UND VERFAHREN ZUM ERZEUGEN EINES BENUTZERDEFINIERBAREN,
VIDEODARGESTELLTEN DOKUMENTS, DAS ANDERUNGEN VON ECHTZEITDATEN ANZEIGT
DISPOSITIF ET PROCEDE PERMETTANT DE CREER UN DOCUMENT AFFICHE SUR ECRAN
VIDEO QUI EST DEFINI PAR L'UTILISATEUR ET QUI PRESENTE LES
MODIFICATIONS APPORTEES A DES

PATENT ASSIGNEE:

TEKNEKRON SOFTWARE SYSTEMS, INC., (1187650), 530 Lytton Avenue, Suite 301
, Palo Alto, California 94301, (US), (applicant designated states:
AT;BE;CH;DE;DK;ES;FR;GB;GR;IT;LI;LU;MC;NL;SE)

INVENTOR:

RISBERG, Jeffrey, Scott, 3249 Morris Drive, Palo Alto, CA 94303, (US)
SKEEN, Marion, Dale, 3826 Magnolia Drive, Palo Alto, CA 94306, (US)
BOWLES MARK, 30 Tripp Court, Woodside, California 94062, (US)

LEGAL REPRESENTATIVE:

Dupuis-Latour, Dominique et al (152552), Avocat a la Cour, Cabinet
Bardehle, Pagenberg & Partner, 45, avenue Montaigne, 75008 Paris, (FR)

PATENT (CC, No, Kind, Date): EP 564548 A1 931013 (Basic)

EP 564548 A1 931229

EP 564548 B1 970917

WO 9212488 920723

APPLICATION (CC, No, Date): EP 92902761 911220; WO 91US9811 911220

PRIORITY (CC, No, Date): US 636044 901228

DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IT; LI; LU; MC; NL;
SE

INTERNATIONAL PATENT CLASS: G06F-017/60 ; G06F-003/033 ; G06F-009/44

CITED PATENTS (EP A): US 4868866 A; GB 2161003 A

CITED PATENTS (WO A): US 4642790 A; US 4750135 A; US 4845645 A; US 4943866
A; US 5058185 A

CITED REFERENCES (EP A):

COMPUTER DESIGN. vol. 29, no. 15 , 1 August 1990 , TULSA, OK, US pages 78

- 94 T. WILLIAMS 'Graphics interfaces make knobs and switches obsolete'

IEEE EXPERT. vol. 3, no. 3 , Fall 1988, , NEW YORK, NY, US pages 18 - 31

D. LEINWEBER 'Knowledge-Based Systems for Financial Applications'

WALL STREET COMPUTER REVIEW. vol. 4, no. 2 , November 1986 , US pages 50

- 56 G. CHRISTOPOULOS 'The Electronic Arsenal of Program Traders'

CANADIAN BUSINESS vol. 57, no. 9 , September 1984 , CA pages 138 - 142 M.

ASNER 'Software Stockpickers'

THE COMPLETE HYPERCARD HANDBOOK. 1988 , NEW YORK, NY, US pages 161 - 183

D. GOODMAN 'All About Buttons'

HUMAN FACTORS IN COMPUTING SYSTEMS, PROCEEDINGS OF CHI'90 April 1990

pages 175 - 182 A. MACLEAN ET AL. 'User-Tailorable Systems: Pressing
the Issues with Buttons'

COMPUTER. vol. 23, no. 11 , November 1990 , LOS ALAMITOS, CA, US pages 71

- 85 B.A. MYERS ET AL. 'GARNET, Comprehensive Support for Graphical,
Highly Interactive User Interfaces'

IBM SYSTEMS JOURNAL. vol. 29, no. 1 , 1990 , ARMONK, NEW YORK, US pages

44 - 58 S.M. FRANKLIN & A.M. PETERS 'Effective application development
for Presentation Manager programs';

NOTE:

No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse: 020619 B1 Date of lapse of European Patent in a
contracting state (Country, date): BE
19970917, DK 19970917, ES 19970917, GR
19970917, IT 19970917, LU 19971231,

Lapse: 20000202 B1 Date of lapse of European Patent in a
contracting state (Country, date): BE
19970917, DK 19970917, GR 19970917, IT
19970917, LU 19971231,

Application: 931013 A1 Published application (A1with Search Report
;A2without Search Report)

Lapse: 20000216 B1 Date of lapse of European Patent in a
contracting state (Country, date): BE
19970917, DK 19970917, GR 19970917, IT
19970917, LU 19971231,

Examination: 931013 A1 Date of filing of request for examination:
930504

Search Report: 931229 A1 Drawing up of a supplementary European search report: 931112
 Examination: 951011 A1 Date of despatch of first examination report: 950825
 Change: 960731 A1 Inventor (change)
 Grant: 970917 B1 Granted patent
 Lapse: 980520 B1 Date of lapse of the European patent in a Contracting State: BE 970917
 Lapse: 980722 B1 Date of lapse of the European patent in a Contracting State: BE 970917, DK 970917
 Oppn None: 980909 B1 No opposition filed
 Lapse: 991020 B1 Date of lapse of European Patent in a contracting state (Country, date): BE 19970917, DK 19970917, IT 19970917,

LANGUAGE (Publication,Procedural,Application): English; English; English
 FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9709W2	4999
CLAIMS B	(German)	9709W2	4726
CLAIMS B	(French)	9709W2	5724
SPEC B	(English)	9709W2	20481
Total word count - document A			0
Total word count - document B			35930
Total word count - documents A + B			35930

INTERNATIONAL PATENT CLASS: G06F-017/60 ...

... G06F-003/033 ...

... G06F-009/44

...SPECIFICATION the Active Objects which are used to compose an active document are as follows.

A label Active **Object** is just a fixed character string placed in a position on a sheet entered by the user. Labels are used to identify sheets, regions on sheets, and individual monitoring Active **Objects** as well as in **script** files to **generate** messages when an alarm **event** occurs or to **generate** customized help screen for a particular active document. A label does not change in real time. Its **attributes** are: String (**field**) which is the text string to be displayed in the label **object** ; and, Alignment (radio button list) which is one of three formatting options - left, center and right.

A quote **object** is a familiar market price quotation. Any number of quotes may be placed on a sheet. **Attributes** of the particular instance record for a quote Active **Object** determine not only what issue is quoted, but also what items of information from the total comprehensive record for a particular issue are displayed. **Different** types of quotes may have **different** formats. The **attributes** of a quote are:

* Market Type (list)

Defines the type of information being received from the source...by the users.

*INTRODUCTION

Features

Features of the MarketSheet(TM) application include:

* Display

The display consists of **objects** on sheets. The user has complete control over the appearance of each **object** . All **objects** have characteristics such as foreground color, background color, font, and border which can be changed to suit...

...drive MarketSheet(TM).

* Dialog Boxes

Pop-up dialog boxes are used to input characteristics associated with each **object** or sheet, such as market symbol, exchange, graph axis limits, etc. Dialog boxes are also used to...

...arranging items on a sheet, such as user-specifiable grids for alignment, and commands to bring overlapped **objects** to the front or to the back. **Objects** can be aligned to the grid either on creation or at any later time. The grid can be adjusted through the sheet edit dialog box, and the grid size is **maintained separately** for each sheet. Layout commands are consistent among the different tools.

* **Scripts**

The user can define a sequence of actions to be carried out when a button is pressed...

...alerts (such as upper and lower limits) in each quote or dynamic graph. The alerts can execute **scripts** in order to change the appearance of monitor items on the screen. The **scripts** can even run other programs, such as a program to play a sound file for audio alerts, or to log the alert in a file.

* **Display Styles**

Each Quote and Ticker **object** uses a display style to format its output. These display styles indicate which fields to show (symbol, price, bid, ask, etc.), how wide to **make** each field, and what colors to use. The flexibility of the style facility allows formats...

...Administrator's Manual describes how to edit the styles and add new ones.

* **Files**

The user can **create** multiple MarketSheet(TM) files that can be saved and reopened, much like the files in the RealTime Spreadsheet. Each file can contain many sheets.

OBJECTS AND TOOLS

The application displays "sheets" or market information. Users can create any number of sheets, which...

25/5,K/16 (Item 16 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00431289

Incremental compiler for source code development system.

Inkrementeller Compiler für ein Quellcodeentwicklungssystem.

Compilateur incrémentiel pour un système de développement de code source.

PATENT ASSIGNEE:

DIGITAL EQUIPMENT CORPORATION, (313081), 111 Powdermill Road, Maynard
Massachusetts 01754-1418, (US), (applicant designated states:
AT;BE;CH;DE;DK;ES;FR;GB;GR;IT;LI;LU;NL;SE)

INVENTOR:

McKeeman, William M., 62 Truell Road, Hollis, New Hampshire, 03049, (US)
Aki, Shota, 9 Grandview Drive, Weare, New Hampshire, 03281, (US)

LEGAL REPRESENTATIVE:

Hale, Peter et al (60281), Kilburn & Strobe 30 John Street, London WC1N
2DD, (GB)

PATENT (CC, No, Kind, Date): EP 406028 A2 910102 (Basic)
EP 406028 A3 930107

APPLICATION (CC, No, Date): EP 90307228 900702;

PRIORITY (CC, No, Date): US 375397 890630; US 375398 890630; US 375399
890630; US 375383 890630; US 375384 890630; US 375401 890630; US 375402
890630

DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IT; LI; LU; NL; SE

INTERNATIONAL PATENT CLASS: G06F-009/46 ; G06F-009/44

CITED PATENTS (EP A): FR 2533721 A; FR 2533721 A

CITED REFERENCES (EP A):

- IEEE SOFTWARE, vol. 4, no. 6, November 1987, pages 36-44, IEEE, New York, US; J.H. WALKER et al.: "The symbolics genera programming environment"
- IDEM
- IDEM
- IDEM
- SIGPLAN NOTICES. PROC. OF THE ACM SIGSOFT/SIGPLAN SOFTWARE ENGINEERING SYMPOSIUM ON PRACTICAL SOFTWARE DEVELOPMENT ENVIRONMENTS, vol. 19, no. 5, May 1984, pages 113-123, Pittsburgh, US; P. FRITZSON: "Preliminary experience from the DICE system a distributed incremental compiling environment"
- IDEM
- IDEM
- JOURNAL OF PASCAL, ADA, & MODULA-2, vol. 4, no. 3, May/June 1985, pages 9-22; R.W. SEBESTA: "Conversational programming systems"
- IDEM
- PROC. OF AN INTERNATIONAL WORKSHOP ADVANCED PROGRAMMING ENVIRONMENTS, 16th June 1986, pages 86-96, Trondheim, Norway; E. ADAMS et al.: "SUNPRO engineering a practical program development environment"
- IBM JOURNAL OF RESEARCH AND DEVELOPMENT, vol. 28, no. 1, January 1984, pages 60-72, Armonk, New York, US; C.N. ALBERGA et al.: "A program development tool"
- IDEM
- IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. SE-11, no. 3, March 1985, pages 276-285, New York, US; S.P. REISS: "PECAN: Program development systems that support multiple views"
- IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. SE-11, no. 3, March 1985, pages 276-285, New York, US; S.P. REISS: "PECAN: Program development systems that support multiple views"
- IBM JOURNAL OF RESEARCH AND DEVELOPMENT, vol. 28, no. 1, January 1984, pages 60-72, Armonk, New York, US; C.N. ALBERGA et al.: "A program development tool"
- IDEM
- IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. SE-11, no. 3, March 1985, pages 276-285, New York, US; S.P. REISS: "PECAN: Program development systems that support multiple views";

ABSTRACT EP 406028 A2

A computer-aided software development system includes programs to implement edit, compile, link and run sequences, all from memory, at very high speed. The compiler and linker operate on an incremental basis, line-by-line or increment-by-increment, so if only one line is changed in an edit session, then only that line and lines related to it need be recompiled if no other code is affected; linking is also done in a manner to save and re-use parts of link tables and link lists which are not changed from one edit session to another. Dependency analysis may be performed incrementally, without requiring the user to enter dependencies. Scanning is also done incrementally, and the resulting token lists and token tables, and lexical increment tables, are saved in memory to be used again where no changes are made. All of the linking tables are saved in memory so there is no need to generate link tables for increments of code where no changes in links are needed. The parser is able to skip lines or blocks of lines of source code which haven't been changed; for this purpose, each line of source text in the editor has a change-tag to indicate whether this line has been changed, and from this change-tag information a clean-lines table is built having a clean-lines indication for each line of source code, indicating how many clean lines follow the present line. All of the source code text modules, the token lists, symbol tables, code tables and related data saved from one compile to another are maintained in virtual memory rather than in files so that speed of operation is enhanced. Also, the object code created is maintained in memory rather than in a file, and executed from this memory image, to reduce delays. A virtual memory management arrangement for the system assures that all of the needed data modules and code is present in real memory in the phase being implemented, with a minimum of page faults and page swapping during the phase, again to enhance operating speed. (see image in original document)

ABSTRACT WORD COUNT: 348

LEGAL STATUS (Type, Pub Date, Kind, Text):

Refusal: 20000301 A2 Date European patent application was refused:
19991009
Application: 910102 A2 Published application (Alwith Search Report
;A2without Search Report)
Examination: 910102 A2 Date of filing of request for examination:
900712
Search Report: 930107 A3 Separate publication of the European or
International search report
Examination: 970108 A2 Date of despatch of first examination report:
961120

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	5467
SPEC A	(English)	EPABF1	14271
Total word count - document A			19738
Total word count - document B			0
Total word count - documents A + B			19738

INTERNATIONAL PATENT CLASS: G06F-009/46 ...

... G06F-009/44

...SPECIFICATION the claims performed by a computer.

In accordance with one embodiment of the present invention, a software **development** system or environment is provided which operates generally on a fine-grain incremental basis, in that increments...

...level. As an example embodiment, a system referred to as a rapid computer assisted software engineering and **development** system (for which the acronym "RCASE" is used below), is disclosed. This system provides a programming environment and a number of facilities or services **designed** to enhance the speed and **productivity** of software **development** engineers, in particular by substantially decreasing the time required for recompilation and relinking in the edit-compile-link-run cycle common to existing traditional software **development** processes. Several different **features**, as disclosed herein, are directed to achieving these goals. The RCASE programming environment employs a fine grain...

...line-at-a-time) compiler including an incremental scanner, an incremental linker, and incremental module dependency analysis (**make**) facility, and a virtual memory manager to reduce or prevent thrashing; a context saving and switching mechanism, and a checkpoint/restart mechanism, are important **features**. Furthermore, the RCASE system is **designed** to operate with any callable editor, callable compiler, or callable debugger that conforms to various interface requirements. A callable **object** file transformer can be included permitting access to applications prepared outside of the environment. Access to runtime libraries is also provided.

In a programming environment according to the invention, the quality of the **object code** is de-emphasized because the goal of reducing the time between editing and running the program is paramount. To increase the speed of the system, the **object code generated** in the RCASE environment is not optimized, resides in virtual memory, and is used only for testing...

25/5,K/17 (Item 17 from file: 348)
DIALOG(R) File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00313968

Apparatus and method using lockout for synchronization of access to main

memory signal groups in a multiprocessor data processing system
Vorrichtung und Verfahren mit Benutzung von Lockout für
Zugriffssynchronisation auf Hauptspeichersignalgruppen in einem
MultiprozessorDatenverarbeitungssystem
Dispositif et methode utilisant exclusion pour synchroniser l'accès à des
groupes de signaux de la mémoire principale dans un système
multiprocesseur de traitement

PATENT ASSIGNEE:

DIGITAL EQUIPMENT CORPORATION, (313081), 111 Powdermill Road, Maynard
Massachusetts 01754-1418, (US), (applicant designated states:
CH;DE;FR;GB;IT;LI;NL;SE)

INVENTOR:

Cutler, David N., 3451 Evergreen Road, Bellevue Washington 98004, (US)
Orbits, David A., 17028 North East 139th Street, Redmond Washington
98052, (US)
Bhandarkar, Dileep, 3 Lantern Lane, Shrewsbury Massachusetts 01545, (US)
Cardoza, Wayne, 3 Hutchinson Road, Merrimack New Hampshire 03054, (US)
Witek, Richard T., 3 Silver Birch Lane, Littleton Massachusetts 01460,
(US)

LEGAL REPRESENTATIVE:

Goodman, Christopher et al (31122), Eric Potter & Clarkson St. Mary's
Court St. Mary's Gate, Nottingham NG1 1LE, (GB)

PATENT (CC, No, Kind, Date): EP 297895 A2 890104 (Basic)
EP 297895 A3 920122
EP 297895 B1 961016

APPLICATION (CC, No, Date): EP 88305994 880630;

PRIORITY (CC, No, Date): US 69380 870701

DESIGNATED STATES: CH; DE; FR; GB; IT; LI; NL; SE

INTERNATIONAL PATENT CLASS: G06F-009/46 ; G06F-009/44

CITED PATENTS (EP A): EP 130593 A; EP 130593 A; EP 121700 A; EP 211384 A

CITED REFERENCES (EP A):

'MC68020 32-BIT MICROPROCESSOR USER'S MANUAL, SECOND EDITION' 1985 ,
PRENTICE HALL , ENGLEWOOD CLIFFS, US
WESCON TECHNICAL PAPERS. vol. 26, September 1982,
NORTH HOLLYWOOD US pages 1 - 5; K. GUTTAG ET AL.: 'Macrostore firmware
emulation as an alternative to co-processors';

ABSTRACT EP 297895 A2

In a multiprocessor data processing unit, a data element in the main
memory unit, that has system wide significance, can have a requirement
that this data element be altered in a controlled manner. Because other
data processing units can have access to this data element, the
alteration of the data element must be synchronized so the other data
processing units are not in the process of altering the same data element
simultaneously. The present invention includes an instruction that
acquires access to an interlock signal in the main memory unit and
initiates an interlock in the main memory unit, thereby excluding other
data processing units from gaining access to the interlock signal
simultaneously. The instruction causes the data element related to the
interlock signal to be transferred to the data processing unit where the
data element is saved, can be entered in mask apparatus and then have a
quantity added thereto. The altered data element is returned to the main
memory unit location and the main memory interlock signal is released,
thereby completing the instruction. (see image in original document)

ABSTRACT WORD COUNT: 184

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse: 030212 B1 Date of lapse of European Patent in a
contracting state (Country, date): CH
19961016, LI 19961016, NL 19961016, SE
19970116,
Application: 890104 A2 Published application (A1with Search Report
;A2without Search Report)
Search Report: 920122 A3 Separate publication of the European or
International search report
Examination: 920909 A2 Date of filing of request for examination:
920713

Examination: 930804 A2 Date of despatch of first examination report:
930621

Grant: 961016 B1 Granted patent

Lapse: 970924 B1 Date of lapse of the European patent in a
Contracting State: SE 970116

Oppn None: 971008 B1 No opposition filed

Lapse: 980121 B1 Date of lapse of the European patent in a
Contracting State: CH 961016, LI 961016, SE
970116

Lapse: 980121 B1 Date of lapse of the European patent in a
Contracting State: CH 961016, LI 961016, SE
970116

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	1138
CLAIMS B	(English)	EPAB96	2533
CLAIMS B	(German)	EPAB96	2095
CLAIMS B	(French)	EPAB96	2954
SPEC A	(English)	EPABF1	3541
SPEC B	(English)	EPAB96	3587
Total word count - document A			4679
Total word count - document B			11169
Total word count - documents A + B			15848

INTERNATIONAL PATENT CLASS: G06F-009/46 ...

... G06F-009/44

...SPECIFICATION a mechanism whereby the change of data elements is synchronized for a plurality of programs by an **instruction** that 1. interlocks access to the memory, 2. reads the logic signal group in the designated memory...

...location, 4. forms the logical product (i.e., the AND function) of the data elements from the **designate** memory location with a mask operand, 5. adds the logical **product** resulting from the operation to an addend operand, 6. writes the resultant sum back to the **designated** memory location and 7. releases the interlock. This **instruction** provides that no other central processing unit or supplementary processor (e.g., an input/output processor) in...

...multiprocessor system can simultaneously be updating the same memory location by means of another interlock and update **instruction**. The **instruction** and apparatus described herein can be used to implement the so-called spin locks, test and set operations, loop iteration induction variable assignment, reference counts on shared **objects** and other operations requiring multiprocessor synchronization. With the use of an EPICODE mode, or where available, a microcoded implementation, this complex sequence of **instructions** can be implemented in an atomic fashion. In addition, by executing the **instruction** in EPICODE mode, additional registers are available and certain **events** are disabled from interrupting execution of the **instruction**. It will be clear that although the RMAQI **instruction** is identified with a quadword data element at the virtual address identified by the **instruction** signal, data elements of any size can utilize the technique of the present invention. In the preferred embodiment, for example, a RMAI **instruction** is provided in which the data processed by the instruction is a longword at the virtual address identified by the instruction. In addition, the preferred embodiment **includes** a RMAQIP instruction, the instruction identifying a physical address. All the instructions are executed in the EPICODE...

...SPECIFICATION a mechanism whereby the change of data elements is synchronized for a plurality of programs by an **instruction** that 1. interlocks access to the memory, 2. reads the logic signal group in the designated memory...

...location, 4. forms the logical product (i.e., the AND function) of the data elements from the **designate** memory location with a mask operand, 5. adds the logical **product** resulting from the operation to an addend operand, 6. writes the resultant sum back to the **designated** memory location and 7. releases the interlock. This **instruction** provides that no other central processing unit or supplementary processor (e.g., an input/output processor) in...

...multiprocessor system can simultaneously be updating the same memory location by means of another interlock and update **instruction**. The **instruction** and apparatus described herein can be used to implement the so-called spin locks, test and set operations, loop iteration induction variable assignment, reference counts on shared **objects** and other operations requiring multiprocessor synchronization. With the use of an EPICODE mode, or where available, a microcoded implementation, this complex sequence of **instructions** can be implemented in an atomic fashion. In addition, by executing the **instruction** in EPICODE mode, additional registers are available and certain **events** are disabled from interrupting execution of the **instruction**. It will be clear that although the RMAQI **instruction** is identified with a quadword data element at the virtual address identified by the **instruction** signal, data elements of any size can utilize the technique of the present invention. In the preferred embodiment, for example, a RMALI **instruction** is provided in which the data processed by the instruction is a longword at the virtual address identified by the instruction. In addition, the preferred embodiment includes a RMAQIP instruction, the instruction identifying a physical address. All the instructions are executed in the EPICODE...

25/5,K/48 (Item 31 from file: 349)
DIALOG(R) File 349:PCT FULLTEXT
(c) 2003 WIPO/Univentio. All rts. reserv.

00300850 **Image available**

UPDATE MECHANISM FOR COMPUTER STORAGE CONTAINER MANAGER

**MOYEN DE MISE A JOUR POUR MODULE DE GESTION D'ELEMENTS DE STOCKAGE
D'ORDINATEURS**

Patent Applicant/Assignee:

APPLE COMPUTER INC,

Inventor(s):

HARRIS Jared M,
RUBEN Ira L,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9519001 A1 19950713

Application: WO 95US196 19950104 (PCT/WO US9500196)

Priority Application: US 94177853 19940105

Designated States: AM AT AU BB BG BR BY CA CH CN CZ DE DK EE ES FI GB GE HU

JP KE KG KP KR KZ LK LR LT LU LV MD MG MN MW MX NL NO NZ PL PT RO RU SD

SE SI SK TJ TT UA UZ VN KE MW SD SZ AT BE CH DE DK ES FR GB GR IE IT LU

MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG

Main International Patent Class: **G06F-009/44**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 119635

English Abstract

Methods and data structures are defined which permit information to be stored as objects in target containers and update containers. A target container defines a first state of the information, and the update container, which can point to the target container, identifies changes to the information in the first state which would be sufficient to update the first information state to a second information state. Update containers may be nested to any depth. When an application program opens an update container, the procedure searches down the chain until it finds the ultimate target container. It then creates in-memory structures for providing access to the objects and value data represented in such container. The procedure then works its way back up the chain, performing the changes on the in-memory structure, which are called for in each of the update containers.

French Abstract

L'invention se rapporte a des structures de donnees et a des procedes permettant de stocker des informations sous forme d'objets dans des elements de stockage cibles et dans des elements de stockage de mise a jour. Un element de stockage cible definit un premier etat relatif aux informations, et l'element de stockage de mise a jour, qui peut identifier l'element de stockage cible, identifie des modifications des informations presentant le premier etat, qui devraient permettre la mise a jour dudit premier etat en un second etat. Les elements de stockage de mise a jour peuvent s'emboiter indefiniment. Lorsqu'un programme d'application ouvre un element de stockage de mise a jour, la procedure appliquee consiste a effectuer une recherche le long de la chaine jusqu'a ce que l'element de stockage cible au bout de la chaine soit identifie. Des structures en memoire sont alors creees afin de permettre l'accès aux objets et aux donnees de valeur representes dans un tel element de stockage. La procedure consiste alors a remonter la chaine, et a effectuer, dans la structure en memoire, les modifications requises dans chacun des elements de stockage de mise a jour.

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Claims

Claim

... additional values that they do not understand; these can be ignored. This allows other applications to annotate **objects** with additional values that may not be generally understood. Typically, these values will be associated with properties...conflicts can occur,

Note that globally unique names are not limited to property and type descriptions. Any **object** can be given a unique name using exactly the same mechanism, and such **object** names may be useful in some applications.

Note also that **objects** can be given short names that are only locally unique, as in the RIFF TOC. These would be a **different** type than Globally Unique Names,

Recall that type and **property** descriptions are **objects** as well. Since types and properties need to have globally unique names, so that applications can recognize them, type and **property** descriptions will typically have a globally unique name value. In many cases, this may be the only contents of a type or **property** description **object**.

In some situations, however, we may wish to put more information into a description. Here are some examples of useful information that can be attached to types or **properties**:

Base types, As previously mentioned, base types allow inheritance of semantics from other existing types for composition into more complex types. Such base type information is intended to include **uses** such as encryption, compression, I/O redirection, etc.

Encoding information. A type definition may indicate the default...type that has a

"UseValue11 and a "NewValue11 handler, Fig, 5

illustrates the structure of in-memory **objects** which are created by the dynamic value mechanism.

Referring to Fig, 5, it is assumed that one...

...with it, indicated by value headers 506 and 508 (424 in Fig. 4). These values are of **different** types, as will be seen from the fact that **different** dynamic value chains are created for these values.

Property 504 also has values associated with it, but these are shown only in the abbreviated form of...entry to point to 11from11 "removed" entry

E: move "from" touched list ("inserted") entry to "to" (original) **object**

delete "removed" entry in "to" touched list

remove "inserted" flag from "from" touched list

entry

if (touched...

...when a value is moved to or within

the same original object. This can be from another

object (moved there'from the original **object** by an earlier move) within the original **object**, but from a **different property** (again from an earlier move).

State 2 is entered when a value is moved to a **different object** from its original.

The general "gist" of this whole thing is to generate a "removed" entry at original **object**

(i.e., moved to a **different property** in the original **object**), Both "removed" and "inserted" are suppressed when a value is moved back to its original **object** and property,

Note, in the code, an actual state number is not

maintained in the touched list...

...01P, to 02P2 as indicated by arrow 1216. There is a "removed" entry 1212 in the source **object**'s touched list and an "inserted" entry 1214 in the destination. The back pointer 1218 is set...values are freed, As an example, if only the values for one property were touched in an **object**, all that would remain in the object's touched list would be a single "deleted property" entry...processed. As mentioned above, each object has its own updating lists for all the values of that **object** in the special updating property value. When the **object** is accessed at open time, all such updates will be applied, This is done to allow for the possibility of other **implementations** of the Container Manager API using random access techniques directly to the container rather than an all in-memory TOC **implementation** described here.

Remember that when an old value is moved, a "removed" touched list entry is **created** at the original source, and an "inserted" entry at the destination. Thus a random access **implementation** could see the source and destination in either order. If the "removed" is seen first, it can do that, The "insert" simply accesses the old value from the container when the destination **object** is processed. Conversely, if the "insert" ...can do that. Then when the "remove" is seen the value is just not added to the **object**. In the present **implementation**, where the entire TOC is loaded into memory, the "removed" entries are ignored. At open time the "insert"

is treated as a move operation. (4) . The "deleted **objects** and properties" list. As each touched **object** is visited on the touched chain (step (3) above), all touched list entries dealing with the values...

...that needs to be done is to walk the chain again. During this walk, all the deleted **objects** and deleted property touched list entries generate appropriate delete updating instructions. The processed touched list entries are...immediately follows. NewType implies only a T follows. This can happen when sequences of touches for an **object** all refer to the same property. This is not the absolute minimum generation, however, since touch list...

...to delete objects and properties accessed as value data from a special iproperty of TOC #1. These **instructions** are **generated** in the following general sequences:

<del obj> 0 <end>

I<del prop1> 0 P [<del prop2> PI J, I I

where <del obj> Deleteobject control byte

O,V P **object** and **property** IDs (4 bytes)

<del prop1> DeleteProperty1 control byte

(**property** for another **object**)

<del prop2> DeleteProperty2 control byte

(**property** for the same **object**)

<end> EndUpdates control byte to mark the end of the updating sequence

What this sequence does is **generate** "delete **object**" or "delete **property**" for a new (<del prop1>) or the same (<del prop2>) **object**.

P. Updating **Instruction** Handlers

The handlers for writing and reading the above

update **instructions** are the same handlers used for TOC I/O, These handlers don't actually do any I...data for such a property are all the updating instructions needed to bring all the values the **associated** object "up-to date". These instructions cannot be applied until the non-private TOC is fully read...

...used

here,

(6). Apply the value updates.

A touched chain was built in step (5) of all

objects containing the special "updating" **property**, The mechanism used to build the chain is identical to that used when recording new updates. The...

...in the updating CCB, After reading in the non-private TOC, the touched chain will represent all **objects** needing updating.

The touched chain can now be walked much like close-time processing to process the updating **instructions** associated with the "updating" **property** of each **object** on the chain. Also like close-time processing, objects on the touched chain are removed from the chain after each updating list is processed. The **object**'s "updating" **property** is also removed, Thus after the chain is walked, the chain is empty and the updating **properties** no longer exist. This puts the in-memory TOC back to its initial state ready to record...me. I in another one of those weird moods today.]

5 PredefTbtPtr P;

TOCObjectPtr theObject;

unsigned short **objectFtags** ;

unsigned long **propertyID**;

0 Boolean attOk = true;

/* Loop through the table to register each type or property...

for (p = predefines; p->**objectID** != 0; ++p) {

if (p->**objectFlags** & Typeobject) { /* if type...

5 **propertyID** = CM StdObjID.GtobatTypeName; /* use std type property ID*/

objectFlags = (TypeObject I Protectedobject); /* these will be the flags*/ else { /* if **property** ... **propertyID** = CM StdObjID

GtobatPropName; /* use std **property** **property** ID*/ 0 **objectFtags** = (Tpropertyo7biect I Protectedobject); /* these will be the fFlags*/ /*

Register (define) the **property** or type...

theObject = cmDefineGtobatNameObject(container, p->**objectID** ,

propertyID,

CM

stdObjID 7BitASCII,

(unsigned c7har *)p->globatName,

container->**generation** , 0. **objectFlags**);

if (altok && theobject == NULL) attOk = false; /* dam!

/* for

0

return (allOk);

/*

I openContainer - common "open a container..."

...control block. For input, the container is opened, the Label read and TOC Loaded. For writing the **container** is simply opened. No matter what the mode, all the other container control block fields are initialized...

handler.cmwriteLabel, CMWriteLbLOpType, RequiredForWriting);

failure 1= buildHandlerVector(container, (CMHandlerAddr *)&container->handler.cmreturnParentValue, CMParentOpType, Optional);

failure 1= buildHandlerVector(**container** , (CMHandlerAddr *)&container->handler.cmreturnContainerName, CMContainerOpName, Optional);

0 failure 1= buildHandlerVector(container, (CHHandlerAddr

*)&container->Phandler.cmreturnTargetType, CMTargetTypeOpType, Optional

```

...
...char *)0xFFFFFFFF && missing != NULL)
    ERROR1(CM-err-UndefRoutine, missing);
    CMfree(missing);
    CMfree(container);
    5 return (NULL);
    /* Set up a object TOC for this container. There are times during
    updating target /* containers that the TOC we're using...

...current container, but for the*/ /* target container. Thus we need two
TOC pointers; one that we usually use for */ /* accessing the "proper"
TOC, and one that always contains the pointer to the TOC we*/ /* create
here for this container. if ((container->privateTOC =
cmCreateTOC(SESSION->cmTocTbtSize, container)) NULL)
    ERROR1(CM.err.NoTOC, CONTAINERNAME...

...point both TOC ptrs are equal*/ 2 0 container->usingTargetTOC = false;
/* ... and there is no target TOC*/ /* Set up the global name symbol
table in much the same way we did the TOC. We need*/ /* two tables here
too. 2 5 if (( container ->privateGlobals = cmCreateGlobalNameTbl(
container )) == NULL)
    ERROR1(CM
err.NoGlobalTable, CONTAINERNAME );
    CMfree( container );
    return (NULL);
    3 0 container ->gtobalWameTable = container ->privateGlobals;
    container ->usingTargetGlobals = false;
    /* Fill in the container control block and append in on to the List of
    open /* containers . At this point, at least, some of the fields which
    are zeroed out w/lt*/ 3 5 /* be...TOC so that we don't release it*/ /*
    prematurely as the closes are done (remember, each updating container
    for mutiple*/ /* updaters will have a copy of this TOC pointer). In
    addition, the "owning" container */ 2 0 /* of the TOC is marked as the
    updating container . This is necessary since we want /* new TOC entries
    (i.e., new updates) to be in the updating container . New TOC*/ /*
    objects always get their owning container from the TOC control block
    whose pointer*/ /* were talking about here. The container , in turn,
    determines which handlers we use .*/ /* By changing the container to
    the updater, we use the updater's handlers. This is*/ 2 5 /* where the
    updaters are to go.
    /* The following diagram illustrates the container /TOC relationships:
    "All updates "BO" updates IOCI
    3 0
    (User's refNum)
    .....
    lContainer Control Btckl --- >lContainer Control...

...RIC"s TOC. Note, however, /* the use count is 3 in this example.
CMCloseContainer0 will close these containers ,*/ /* from l8C11 to "A"...
It will attempt to free the TOC. But the actual freeing of the...

...that we changed the container as far as the Low-Level*/ /* TOC routines
are concerned. They only use it to get at the global session data /*
which is the same (hopefully) no matter which container we use . But
they do put it 65 /* into the object as described above.
container->privateTOC = container->toc; /* save our TOC
container ->usingTargetTOC = true; /* signal we did this*/
container -noe = cmUseTOC(targetContainer->toc, conta iner); /* use
target's TOC*/ /* In a manner similar to the TOC, we must use a common
global name tree. This code /* is exactly the same as for the TOC, but
for the global name symbol table.*/
5 container ->privateGlobals = container ->gtobalNameTable;
container ->usingTargetGiobats = true;
container ->gtobatNameTable =
cmUseGtoba (NameTbt (targetContainer->giobaLNameTable,
container );

```

```

1 0 /* Carry forward the final ("bottom") target container pointer,
i.e., the original*/ /* container before any updates were applied. it
is put in targetContainer. By */ /* knowing the "top" and "bottom".
CMCloseContainero knows how many containers need /* closing. in
addition, since the final target is the same for all the updaters of /*
it (since we are carrying it forward through each updater), the
targetContainer is /* is used for all container consistency checks,
i.e., checks in routines that take /* multiple objects to verify all
the objects came from the "same" container . This*/ /* allows the
objects to be from different containers so long as they are
updaters.*/

```

```

container ->targetContainer = targetContainer->targetContainer;
2 0 3 /* wasn't this fun?*/

```

```

/* ..... :
..... I openTargetContainer - open appended container
(TOCS) and apply their updates 2 5

```

```

*.....
..... This routine is called by both CMOpenContainero and
operkAppendedTargetToBeLJUpdatedo after it has opened a container for
reading or an appended container layer that is to receive new updates.
This routine looks to see if there are (more) appended Layers of 3 0
updates and opens those containers as targets to apply the updates to
bring the targets up-to-date. This function always returns...done in such
a way as to produce a single TOC, but with entries tagged to the
container doing the updating. The updating container is always
returned. In the end the original container that was the "newest"
update (the first one we Looked at) is returned and that is what the user
will use.

```

```

static ContainerPtr CM.NEAR openTargetContainer( ContainerPtr
container ,

```

```

2 0 TOCObjectPtr theTOCObject)
ContainerPtr targetContainer, savedUpdatingContainer;

```

```

TOCPropertyPtr theProperty;
TOCValueNdrPtr theValueNdr, targetValueHdr;

```

```

2 5 CMProperty targetProperty;

```

```

CMType targetType;

```

```

/* We have to use the "pointing value", to get at the target. The
target may be /* appended or separate as determined...

```

```

...pointing value, in particular, its type.*/ 3 0 /* We do a CMUseValue0 on
the pointing value's object , property , and type refNums.*/ /* For a
separate container a dynamic value will be spawned. This is discussed
*/ /* shortly. But first, we must get the refNums. Just to be paranoid,
we check that*/ /* we indeed get this stuff. 3 5 targetProperty
=(CMProperty)cmFindObject( container ->toc, CM.StdObjID.TOC-Target);
theProperty = cmGetObjectProperty(theTOCObject, CM.StdObjID.TOC-Target);
if (theProperty) {

```

```

4 0 targetValueHdr...updatingContainer points to itself, so all the
Lower*/ /* Level target's updatingContainer will point there as well.

```

```

container ->openingTarget = true; /* target gets our updatingContainer*/
2 0 targetContainer =

```

```

(ContainerPtr)CMOpenContainer((CMSession)container->sessionoata,
(CMRefCon)theValueHdr,
(CMGLobName...itself, although we read them both with this single read.
The first subsection are 5 0 /* "new property (values) for old objects
". The second subsection is just new objects..*/ /* One of the "new
properties for old objects " will be a special "updating" property ,*/
/* whose value points to the updating instructions on how to update all
the values of*/ /* its associated object . These instructions cannot be
applied until the non-private */ /* TOC is fully read in because there
could be forward reference to other objects . So*/ /* if the special
"updating" property is spotted for any object , that object is
placed*/ /* on the "touched chain" whose real use is for remembering who
got touched during*/ /* NEW...

```

```

...first" with respect to whose /* container we're talking about and from
whose container we thread the touched /* objects that are to receive
those updates (which container "owns" the touched /* chain). /* ALL

```

generation of TOC information and all the updating manipulations are set up for*/ /* "normal" processing, i.e., when a container is fully open and new updates are being...

...to the */ /* user. It acts as an "anchor point"; a place to look at no matter which **container** */ /* pointer we happen to have. Remember, connectTargetToToltsUpdater, just called /* above, connects the containers so they are all using the SAME TOC. TOC entries /* thus will be "tagged" with the **container** that owns them. The updatingContainer*/ /* pointer is sent "down" as each **container** is opened so that they immediately have /* the "proper" **container** to point to. The problem, and subtle point, is with that
/* word, "proper"!
/* To "tag" each new TOC entry with its owning **container**, all entry creation use the /* updatingContainer. In the normal case, that's what we want; the **container** that is /* to receive all new values. For creating a new updating **container**, that would be /* the top-most **container** we return to the user. /* When recording updates, all touched objects are chained to the touched chain. The*/ 2 0 /* "owner" of this chain is again the updatingContainer. /* Thus, updatingContainer is the key pointer controlling value tagging and **object** /* chaining. At TOC Load time the updatingContainer must be the one where new values*/ /* are to go and where **objects** needing updating are to be chained to. The subtle 5 /* point here (finally) is that updatingContainer is pointing at the TOP MOST*/ /* **container** and, HERE, as we're unwinding the recursive opens to insert new TOC*/ /* entries and to chain **objects** needing updating, the top-most TOC is NOT what we
/* want!
0 /* The container of interest here is the CURRENT container. This is the one just*/ /* **connected** to its target (no matter how many it was previously **connected** to -- we*/ /* view it here as just one). This is the one we're about to load...

...debug! it may be obvious now. It wasn't

0 /* then! This was one killer bug!

if ((**container** ->useFlags & kCMWriting) == 0) /* if not new updating container if (container->tocNewValuesValue == NULL) /* ... there must be updating TOC...

...updatingContainer*/

/* Load in the non-private TOC of this container and build the touched chain of all*/ /* **objects** needing updating. if (TOCNewValuesTOCSize > 0) /* if new value updates exist ...

container->tocFuttyReadIn false; (safety status switch)*/

if (IcmReadTOC(**container**, TOCNewValuesTOCOffset, TOCNewValuesTOCSize))

UndoOpen;

return (NULL);

container ->tocFuttyReadIn = true; /* reset safety switch

#if CMDUMPTOC /* show TOC in **container** */

0 if (SESSION->cmDbgFile) /* ... if debugging

if (**container** ->majorVersion > 1) /* ... and not format I TOC*/

CMDumpContainerTOC((CMContainer) **container**, SESSION->cmDbgFile,

TOCNewValuesTOCOffset, TOCNewValuesTOCSize);

#endif

/* Now **apply** the updates in this **container** to the target **container**. The touched /* chain built during the reading in of the ...TOC is walked and processed.*/ /* There is also a separate chunk of updates that deal solely with **object** and */ 2 0 /* **property** deletions. These are pointed to by a special "deletes" **property** in the*/ /* private TOC M. /* Remember, that we're **applying** all these updates as we unwind the recursion, i.e.,*/ /* after CMOpenContainer called above returns. The recursion...

...corresponding to the specified typeName is opened for input or for updating by reusing free space. The **association** between the typeName and the physical container is through the metaHandler defined for that same typeName. The...field valid anyway.

theTOCObject = cmFindObject(container->toc, CM

```

StdobjID
TOC);
if (theTOCObject == NULL) f /* error if TOC object is missing*/
ERRORUCM err MissingTOCObj, CONTAINERNAME);
UndoOpen(5;
return (NULL);
0 theTOCObject->objectFlags I= Protectedobject; /* this object...

...CONTAINERNAME);
Undoopen(@; @
return (NULL);
2 5 theValueHdr->valueFlags I= ValueProtected; /* don't allow writing to
this value*/
    container -> generation = theSeedValue->theValueHdr->Seneration;
    container ->nextUserObjectID = theSeedValue->vatie.im.uLongValue;
3 0 container ->tocIDSeedVatie = theSeedValue; /* save ptr to the seed
value hdr*/ /* The minimum seed value is used when applying updates to
a target container to */ /* suppress objects in the updating
container with IDs less than the min from being */ applied to the
target. Such objects can occur during the opening of new updating 3 5
/* containers before we can get at the target to know what the seed
should be.*/
theProperty = cmGetObjectProperty(theTOCObject...

...if (theProperty) (
theValueHdr = (TOCVatieHdrPtr)cmGetListHead(&theProperty->vatieHdrList);
if (theValueHdr && theValueHdr->typeID == CM StdObjID TOC Type)
4 0 container ->tocIDMinSeedVatie =
(TOCVatiePt@r)cmGetLiitHe7ad(&theValueHdr->valueList);
else
theVatieHdr NULL;
if (theProperty NULL 11 theValueHdr == NULL 11 container...

...writing to this value*/ /* The size property is used to get at the size
in the TOC object . This mist be in /* agreement with the TOC size in
the Label which we already extracted to...

```

25/5,K/49 (Item 32 from file: 349)
 DIALOG(R)File 349:PCT FULLTEXT
 (c) 2003 WIPO/Univentio. All rts. reserv.

00254739

USE OF A LANGUAGE WITH A BUILT-IN INTERPRETER FOR PRODUCING A USER
 INTERFACE DEFINING INTERACTIVE TOOL
 UTILISATION D'UN LANGAGE A INTERPRETE EMBARQUE POUR LA REALISATION D'UN
 OUTIL INTERACTIF DE DEFINITION D'INTERFACE UTILISATEURS

Patent Applicant/Assignee:

BULL S A,
 NAHABOO Colas,
 BOUTHORS Vincent,

Inventor(s):

NAHABOO Colas,
 BOUTHORS Vincent,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9402895 A1 19940203
 Application: WO 93FR749 19930721 (PCT/WO FR9300749)
 Priority Application: FR 929057 19920722

Designated States: JP KR US

Main International Patent Class: G06F-009/44

Publication Language: French

Fulltext Availability:

Detailed Description
 Claims

Fulltext Word Count: 16736

English Abstract

An interactive interface-defining tool using an interpreted language with a similar representation for data and programs, and an interpreter included in the interface definition program consisting of a mixture of "C" language instructions and interpretable language instructions which co-operates with a library of "X/MOTIF" control interactive objects ("widgets") and a library of graphical objects "GO".

French Abstract

La presente demande concerne un outil interactif de definition d'interface utilisant un langage interprete ayant une representation similaire pour les donnees et les programmes et a interpreteur embarque avec le programme de definition d'interface constitue d'un melange d'instructions en langage "C" et d'instructions en langage interpretable qui coopere avec une bibliotheque d'objets interactifs "widget" de commande "X/MOTIF" et une bibliotheque d'objets graphiques "GO".

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Detailed Description

Detailed Description

... super-class [El 1 oved ... @ returns superclass
eg: cast [El] 1. 1 change the type of an **object**
type-is-subtype [W21 r7l tests if a type is subtype of
another
6 2 **Attribute**
Function Action
eg: **attribute** -atom [El] ZI returns information about a widget
attribute
eg: **attribute** - make [El] n? defines a widget **attribute**
eg: **attribute** -name [El] M returns information about a widget
attribute
eg: **attribute** -type [El] M returns information about a widget
attribute
6.2,3 Behavior
Function Action
XtAddCallback [El] adds or remove callback in a widget
XtAddEventHandler [El] add or remove an **event** handler
XtRemoveCallback [El] C? adds or remove callback in a widget
XtRemoveEventHandler [El] M add or remove...

25/5,K/50 (Item 33 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00253666 **Image available**

SYSTEM FOR IMPLEMENTATION-INDEPENDENT INTERFACE SPECIFICATION
SYSTEME DE SPECIFICATION D'INTERFACE INDEPENDANTE DE LA VERSION

Patent Applicant/Assignee:

TELEFONAKTIEBOLAGET LM ERICSSON,

Inventor(s):

LUNDIN Lars Kenneth,

WIMAN Lars-Erik,

SVENSSON Mats Ragnar,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9401820 A1 19940120

Application: WO 93SE458 19930525 (PCT/WO SE9300458)

Priority Application: US 92293 19920701

Designated States: AU BR FI KR NO AT BE CH DE DK ES FR GB GR IE IT LU MC NL
PT SE

Main International Patent Class: **G06F-009/44**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 10134

English Abstract

A system for facilitating communications between components, or parties, of a computer system. The system has a high level interface specification (172) which includes a communications contract between pairs of communicating parties. The interface specification (172) is input into a generation tool (170) to generate party specific implementation of the interface specification (172). The generated party specific implementation is then used to permit operative communications between parties.

French Abstract

Le systeme decrit facilite les communications entre des composantes, ou des correspondants, d'un systeme informatique. Le systeme possede une specification d'interface de haut niveau (172) qui comporte un contrat de communication entre des couples de correspondants en communication. La specification d'interface (172) est introduite dans un outil de creation (170) pour creer une version, propre a des correspondants, de la specification d'interface (172). Cette version est ensuite utilisee pour permettre des communications fonctionnelles entre des correspondants.

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Detailed Description

Detailed Description

... by means of the code residing in the server software located in the "server" software unit 130.

Objects are instances of classes which are language constructs that contain both data and **code** or functions within a single package or unit. Because they are able to contain definitions of both...

...or unit, they act as miniature, independent programs. They can be used, therefore, as building blocks in **creating** more complex programs without having to redevelop the **code** necessary for those functions. Because they can be **maintained** and modified **independently**, program maintenance and revision is simplified.

A class is a template that is used to define an...

25/5,K/51 (Item 34 from file: 349)
DIALOG(R) File 349:PCT FULLTEXT
(c) 2003 WIPO/Univentio. All rts. reserv.

00234265 **Image available**

SYSTEM FOR DIVIDING PROCESSING TASKS INTO SIGNAL PROCESSOR AND
DECISION-MAKING MICROPROCESSOR INTERFACING
SYSTEME DE SEPARATION DES TACHES DE TRAITEMENT EN TACHES POUR INTERFACAGE
AVEC UN PROCESSEUR DE SIGNAUX ET UN MICROPROCESSEUR DE PRISE DE
DECISION

Patent Applicant/Assignee:

STAR SEMICONDUCTOR CORPORATION,

Inventor(s):

ROBINSON Jeffrey I,
ROUSE Keith,
KRASSOWSKI Andrew J,
MONTLICK Terry F,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9308524 A1 19930429

Application: WO 92US8954 19921014 (PCT/WO US9208954)

Priority Application: US 91776161 19911015

Designated States: AU CA JP KR AT BE CH DE DK ES FR GB GR IE IT LU MC NL SE

Main International Patent Class: G06F-009/00
International Patent Class: G06F-09:40
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 219172

English Abstract

Architectures and methods are provided for efficiently dividing a processing task into tasks for a programmable real time signal processor (SPROC) (10) and tasks for a decision-making microprocessor (2120). The SPROC is provided with a non-interrupt structure where data flow is through a multiported central memory. The SPROC is also programmed in an environment which requires nothing more than graphic entry of a block diagram of the user's design. In automatically implementing the block diagram into silicon, the SPROC programming/development environment accounts for and provides software connection and interfaces with a host microprocessor (2120). The programming environment preferably includes: a high-level computer screen entry system which permits choosing, entry, parameterization, and connection of a plurality of functional blocks; a functional block cell library (2015) which provides source code representing the functional blocks; and a signal processor scheduler/compiler (2040) which uses the functional block cell library (2015) and the information entered into the high-level entry system to compile a program and to output source program code for a program memory and source data code for the data memory of the SPROC, as well as a symbol table which provides a memory map which maps SPROC addresses to variable names which the microprocessor (2120) will refer to in separately compiling its program.

French Abstract

On decrit des architectures et procedes qui permettent de separer efficacement une tache de traitement en taches destinees a un processeur de signaux programmable fonctionnant en temps reel (SPROC) (10) et a un microprocesseur de prise de decision (2120). Le SPROC est dote d'une structure depourvue d'interruption ou le flux de donnees arrive par l'intermediaire d'une memoire centrale a ports multiples. Il est aussi programme dans un environnement n'exigeant rien d'autre que l'introduction graphique d'un schema global relatif aux intentions de l'utilisateur. Avec la realisation automatique du schema global dans le silicium, l'environnement de programmation et de developpement du SPROC prend en compte et fournit la connexion au logiciel et realise une interface avec un microprocesseur hote (2120). Cet environnement de programmation comporte de preference un systeme d'introduction a ecran d'affichage perfectionne qui permet de choisir, introduire, parametriser et fournit une connexion avec differents blocs fonctionnels; une bibliotheque a cellules de bloc fonctionnel (2015) qui fournit un code source representant les blocs fonctionnels; et un programmeur/compilateur pour processeur de signal (2040). Ce dernier utilise la bibliotheque a cellules (2015) et l'information introduite dans le systeme d'introduction perfectionne pour compiler un programme et delivrer en sortie un code de programme source concernant une memoire du programme et un code de donnees source destine a la memoire de donnees du SPROC, ainsi qu'une table de symboles qui fournit une cartographie memorisee, contenant les adresses donnees par le SPROC aux differents noms auxquels le microprocesseur (2120) viendra se referer en compilant separement son propre programme.

Main International Patent Class: G06F-009/00
International Patent Class: G06F-09:40
Fulltext Availability:
Claims

Claim

... S1++], START
RECORD);

```

/* put start Lines into SMI file
if (Make.SMI.file) C
strcpy(record, 11/* Generated by makeLand.exe $Revision: 1.15 $ *An11);
strcpy(Out-buf-SMI[Row.SMI++], record);
strcpy(record, 11# incLude
11sprocdef.hX11
n
n11);
strcpy(Out-buf-SM1[Row.SMI++], record);
sprintf(record, "before-const fix24-type after
const %s-bLock[] WI, Design -name);
strcpy(Out-buf-SM1[Row.SMI++], record);
/* handle extra files, if needed
if ( Make -extra-files) C
/* use these fixed names
strcpy(cut- code fileTname, " code ");
strcpy(out-control.file-name, "control");
strcpy(out data file name, "data");
D
SUBSTITUTE SHEET
1* and these flle name extensions I/
apply.fiefaUlt.extension(out. code -file-name@, ext.str);
apply.fiefaUlt.extension(out@.control.file-name, ext-str);
applyjef&uLt.extension(out...

```

25/5,K/52 (Item 35 from file: 349)
 DIALOG(R) File 349:PCT FULLTEXT
 (c) 2003 WIPO/Univentio. All rts. reserv.

00148298 **Image available**

MICROPROGRAMMABLE LANGUAGE EMULATION SYSTEM
SYSTEME MICROPROGRAMMABLE D'EMULATION DE LANGAGES

Patent Applicant/Assignée:

INTERNATIONAL META SYSTEMS INC,

Inventor(s):

SPECKHARD Arthur E,

THAMES Joseph M,

Patent and Priority Information (Country, Number, Date):

Patent: WO 8805190 A1 19880714

Application: WO 87US3444 19871228 (PCT/WO US8703444)

Priority Application: US 87714 19870106

Designated States: DE FR GB IT JP

Main International Patent Class: **G06F-009/34**

International Patent Class: **G06F-09:44**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 14408

English Abstract

A data processing system executes a high level computer language program by encoding statements of the program into variable length tokens and then executing the tokens. Each token is a variable length bit field having a value representative of a semantic element of a program statement and a length representative of the context of the semantic element. A single very large scale integrated circuit (VLSI) processor (40) is microprogrammed to execute the encoded program in a pipeline manner and may be used in conjunction with a host computer (20), such as an IBM PC AT or equivalent system.

French Abstract

Un systeme de traitement de donnees execute un programme d'ordinateur en langage evolue en codant des instructions du programme en des symboles de longueur variable et en executant ensuite les symboles. Chaque symbole est un champ binaire de longueur variable dont la valeur represente un element semantique d'une instruction du programme et dont la longueur

represente le contexte de l'element semantique. Un seul processeur (40) a circuits integres a tres grande echelle (VLSI) est microprogramme pour executer en pipeline le programme code et peut etre utilise en combinaison avec un ordinateur hote (20), tel qu'un IBM AT ou un systeme equivalent.

Main International Patent Class: G06F-009/34

International Patent Class: G06F-09:44

Fulltext Availability:

Claims

Claim

... Variable: (2:2)(4:class)(primary)[(secondary)]

Name: (2:3)(4:3)(4:class)(primary)[(secondary)]

The **second field** of the name reference is actually a type **code** (4:3) referring ...name," since a name is a literal data structure, The reference structure has a four bit class **field**, a variable format primary reference and may have a secondary reference relative to the primary one (common...

...primary reference is an index

to one of the tables in the contour 130 portion of the **object** program, as designated by the class field. The four-bit class field and **associated** class designations are listed in TABLE IV below:

TABLE IV

Class Code Class Designation

{4:0=inv...third operand, The RHS portion of an instruction contains a second operation, including external bus instructions, subroutine **link** /return skips, transfers/ or memory indexing.

A three address instruction format is used for LHS

L& operations; having the following symbolic form:

T := A op B

wherein a register specified by the "IT" **field** is assigned (:=) the value of some binary operation (op) performed on the contents of the register specified by the "All **field** and the contents specified by the IIBII **field**. -Considering the example FOKTRAN assignment. statement of FIG. 4, A = 10 + 5, the T **field** of an LHS **instruction** specifies a register **containing** the local variable "A," the A **field** specifies a register **containing** the value of 10 and the B **field** specifies a register **containing** the value of 5.

The 24 operations **implemented** in the processor are

designated by the Operator (Op) and C **fields**, as shown TABLE IX below:

TABLE IX

Operator **Field**

3G

OP=011 Left Shift A by B (0: rotational, 1 : logical)

Op=213 Addition, A+B...820. The LHS is decoded for the Operator, A and B fields but not for the T **field**, which T CY,

is passed unaltered to the next phase, The A and B fields designate registers...

...which determines which

operation is specified by the Operator field.

The RHS is also decoded during the **second** phase Ph but only the Unconditional Transfer, Link,, Link conditional, Return and Load K Register **instructions** are acted upon during the second phase, All other RHS

1

instructions are passed to the next phase. For the Unconditional Transfer **instruction**, the address field of the **instruction** is used to select the next value for the location

counter 830, which references the address for the **instruction** to be fetched immediately after the **instruction** in Ph I advances. The Link and Link Conditional **instructions** are executed in similar fashion to the Unconditional Transfer **instruction** but in addition they "push" the accompanying location counter value onto the Link Stack Register 900 for subsequent **use** by the Return **instruction**. That **instruction** "Pops" a value from the Link Stack Register 900 and adds the value of its address **field** to the popped value. The sum - is used as the next value of the location counter 830. The Load K Register **instruction** loads its address **field** into the K Register 910, which is a special purpose register used in conjunction with a Memory...

...The values selected in the second phase Ph II are operated upon as specified by the Operator **field** and the result is sent to an X Register 930, which gives the succeeding instruction access to...

Set	Items	Description
S1	13062	OBJECT? OR CUSTOM(1W)CONTROL? OR SOFTWARE()ROUTINE? OR DLL OR DLLS OR CODE()MODULE? OR VBX OR VBXS OR VISUAL()BASIC()EXTENSION? OR READY()MADE() (COMPONENT? OR CONTROL?) OR OCX OR OCXS OR (ACTIVEX OR ACTIVE()X)()CONTROL?
S2	74602	GENERAT? OR MAKE? ? OR MAKING OR CREATE? OR CREATING OR PRODUCE?
S3	16097	SCRIPT? OR INSTRUCTION? OR RULE? OR SYNTAX OR CODE OR CODES
S4	52	(MAINTAIN? OR PRESERVE? OR KEEP? ? OR SUSTAIN? OR SUPPORT? -) (3N) (SEPARATELY OR INDEPENDENTLY OR INDIVIDUALLY OR SINGLY)
S5	42311	WRAP? OR CONTAIN? OR INCLUDE? OR HOLD? OR ENCLOSE?
S6	64567	ADDITIONAL OR ANOTHER OR SECOND OR 2ND OR DIFFERENT OR MORE OR SECONDARY OR SUPPLEMENT? OR OTHER
S7	39470	PROPERTY OR PROPERTIES OR FIELD? OR PATTERN? OR QUALITY OR QUALITIES OR ATTRIBUTE? OR FEATURE?
S8	29005	EVENT? OR HAPPENING? OR INCIDENT? OR OCCURRENCE? OR DEVELOPMENT? OR EPISODE?
S9	3374	ABOVE OR BEYOND OR EXCEED? OR SURPASS? OR OVER()REACH? OR - OVERREACH?
S10	39279	INTERNAL OR CONFIGUR? OR DESIGN? OR ARRANGE? OR SET()UP OR SETUP OR PROGRAM?()IN OR SETTING? OR IMBED? OR EMBED? OR FIXED OR ENTRENCH? OR FASTEN? OR INFIX? OR INGRAIN
S11	50787	UTILIZE OR APPLY? OR APPLIES OR EMPLOY? OR IMPLEMENT? OR USE OR USES?
S12	42257	LINK? OR (PUT OR LAY OR PIECE)()TOGETHER OR JOIN? OR UNITE? OR UNIFY OR UNIFIES OR CONNECT? OR ASSEMBLE? OR COLLECT? OR - COMBINE? OR ASSOCIAT? OR AFFILIAT? OR TIE? OR TRANSACTION?
S13	2543	SEQUENCE? OR QUEUE OR QUEUED OR QUEUING OR QUEUEING OR RANK? OR (LOGICAL OR SERIAL)()ORDER OR CONSECUTIVE?
S14	3061	S1 AND S2 AND S3
S15	848	S5 AND S1 AND (S6 (5N) (S7 OR S8))
S16	5002	(S7 OR S8) AND S10 AND S1
S17	4465	S11 (S) (S6 (5N) (S7 OR S8))
S18	6553	(S12 OR S13) AND S1
S19	1	S14 AND S4
S20	41	S14 AND S15 AND S16 AND S17 AND S18
S21	42	S19 OR S20
S22	9	S21 NOT PY>1994
S23	9	S22 NOT PD>19941116

File 256:SoftBase:Reviews,Companies&Prods. 82-2003/Nov
(c)2003 Info.Sources Inc

23/5/1

DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

01789909 DOCUMENT TYPE: Product

PRODUCT NAME: Topocad 4.1 (789909)

Chaos systems AB (656429)
Solna Strandvag 18, 6th Floor 171 54
Solna, Sweden
TELEPHONE: () 082-90660

RECORD TYPE: Directory

CONTACT: Sales Department

Topocad 4.1 is an efficient and easy-to-use CAD system for surveyors, mappers, and civil engineers. It enables users to work with all types of data, e.g., survey data, coordinate files, or drawings. It provides surveyors with the tools they need to create useful maps and models, without complications or unnecessary functions. Topocad can also output to most filetypes and is available in 14 spoken languages, which makes it ideal for international projects. Topocad simplifies data acquisition in the field through its use of code tables. It can also import a variety of file formats, including DWG, DGN, and DXF. Users can link attributes to codes, symbols, points, and files. Topocad supports an unlimited number of points, objects, codes, and more. It also offers advanced digital terrain modeling (DTM) functions such as height extraction from models. Other features include text mode editing toolbox; object-oriented design; default drawing preferences; instrument and field book communications; road line management functions; Helmer and Affin calculations; global and local line types; and support for all Windows plotters and printers. Topocad 4.1 can be extended with several add-ons such as Project Design, Network Adjustment (for advanced triangulation), and Longitudinal Section.

DESCRIPTORS: CAD; CAD CAM; CAE; Civil Engineering; Data Acquisition;
Foreign Language Packages; Graphics for Science & Engineering; Mapping;
Surveying

HARDWARE: IBM PC & Compatibles; Pentium
OPERATING SYSTEM: Windows; Windows NT/2000; Windows XP
PROGRAM LANGUAGES: OOP (Object Oriented Programming)
TYPE OF PRODUCT: Micro
POTENTIAL USERS: Surveyors, Mapping, Civil Engineers
DATE OF RELEASE: 01/1995
PRICE: \$995; Net Adjustment--\$1,500; most other add-ons \$600; Tunnel
model--\$1,50

NUMBER OF INSTALLATIONS: 1600
DOCUMENTATION AVAILABLE: User manuals; reference manuals
TRAINING AVAILABLE: Technical support; training; e-mail support; fax
support
OTHER REQUIREMENTS: 24MB RAM; Pentium+ CPU; Win 9x+ required
SERVICES AVAILABLE: Consulting
REVISION DATE: 20000530

23/5/2

DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

01751979 DOCUMENT TYPE: Product

PRODUCT NAME: Canvas 8 (751979)

ACD Systems International Inc (699998)
PO Box 36
Saanichton, BC V8M 2C3 Canada
TELEPHONE: (250) 544-6700

RECORD TYPE: Directory

CONTACT: Sales Department

Canvas 8 from ACD Systems International provides cutting-edge tools that allow graphics users to **create** and edit vector drawings, bitmap images, and text in a single application. Canvas users can **create** illustrations, image compositions, and documents without purchasing hundreds of dollars worth of software or spending many hours learning how to **use** multiple applications. Canvas's SpriteLayer (TM) technology supports advanced methods of layering and blending vector drawings, images, and text with control over transparency. Cutaways, vignettes, and collages are easy to accomplish, and users never lose editing **properties** or have to convert to **another object** type. There is no need to learn how to **use** channels or layers. Canvas 8 offers both creative power and precision, robust photo editing tools, and page layout abilities. With Canvas, basic opacity effects are as easy as selecting an **object** or group and choosing a transparency level from 0 to 100 percent. The user can also refine the effect with Transfer Modes, including Multiply, Lighten, Darken, and Screen. Vector transparency tools allow users to **apply** directional, radial, rectangular, or elliptical transparency fades to any **object**, so they can easily build and blend vignettes. Users can simply drag-and-drop an **object** to **apply** the effect. More precise positioning can be achieved by numerically specifying the position and range of transparency. Newer **features** of Canvas **include** a **sequence** recorder/player, a **scripting** engine, and image sharing tools.

DESCRIPTORS: Business Graphics; Color Separation; Draw; Graphics Tools; Image Processing; Presentations; Web Site **Design**

HARDWARE: Apple Macintosh; IBM PC & Compatibles; Pentium; PowerMac
OPERATING SYSTEM: MacOS X; Windows; Windows NT/2000; Windows XP
PROGRAM LANGUAGES: PostScript
TYPE OF PRODUCT: Micro
POTENTIAL USERS: Artists, Business Graphics, Cross Industry
PRICE: Available upon request; \$375--estimated street price; upgrade--\$199; 15-day Internet trial available

REVISION DATE: 20030706

23/5/3

DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

01729787 DOCUMENT TYPE: Product

PRODUCT NAME: WebLOAD 6.0 (729787)

RadView Software Inc (649236)
2 Habarzel St
Tel-Aviv 69710, Israel
TELEPHONE: () 972-37650555

RECORD TYPE: Directory

CONTACT: Sales Department

RadView Software's WebLOAD (TM) 6.0 is an e-business testing and verification tool that allows companies to ensure the efficiency and performance of n- **tier** Web-based applications, business systems, and Web services before actual deployment. WebLOAD simulates realistic Web traffic

using virtual clients, whose behavior is then registered in JavaScript-based test **scripts** . Reporting of the results is done graphically, in real time. WebLOAD tests for scalability and functional integrity under pre-determined load conditions. The Data Drilling function allows users to analyze very specific aspects of functionality, e.g. at the level of an individual client or **transaction** . The Table Compare Viewer graphically displays results of both successful and unsuccessful **transactions** for any defined **event** . Users can **configure** WebLOAD to analyze any type of workload situation according to any parameter or **property** , such as caching, multithreading, **different connection** types, and strength of SSL encryption. WebLOAD's distributed architecture provides high-level load- **generation** and can incorporate additional systems for testing. WebLOAD defines, monitors, and runs tests from a centrally managed console. Integrated with Sun Solaris and Microsoft Windows performance monitors, WebLOAD informs users of critical system data, such as memory consumption and CPU **use** . The Cruise Control module lets Webmasters predict performance limits to get the most **use** from Web applications. **Other features include** multiple IP support; DOM (document **object** model) support; simulation of **connections** of variable speeds; more than 75 performance metrics for accurate and flexible testing; the Visual Agenda Authoring Tool (VAAT), for **creating test scripts** ; **script** sharing; and .NET, SOAP, WSDL, and BPEL4W5 support.

DESCRIPTORS: E-Commerce; Internet Traffic Analysis; Network Administration
; Performance Monitors; Software Testing; System Monitoring; System
Performance; Web Servers; Web Services; Webmasters

HARDWARE: IBM PC & Compatibles; Sun; UNIX

OPERATING SYSTEM: .NET; Linux; Solaris; Sun ONE; WebSphere; Windows
NT/2000; Windows XP

PROGRAM LANGUAGES: Java; JavaScript; WSDL

TYPE OF PRODUCT: Mini; Micro; Workstation

POTENTIAL USERS: Cross Industry, Webmasters, Site Owners, E-Commerce, Web
Services

PRICE: Available upon request; Internet demo available

OTHER REQUIREMENTS: Win 2000+ or Solaris 2.6+ or Red Hat Linux 6.2+
required

REVISION DATE: 20030804

23/5/4

DIALOG(R)File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

01138991 DOCUMENT TYPE: Product

PRODUCT NAME: Fujitsu NetCOBOL for .NET (138991)

Fujitsu Software Technology Corp (Fujitsu-Softek (394572)
1250 E Arques Ave M/S 317
Sunnyvale, CA 94088-3470 United States
TELEPHONE: (408) 746-7638

RECORD TYPE: Directory

CONTACT: Sales Department

Fujitsu Software Technology's Fujitsu NetCOBOL for .NET, offered in Professional, Developer, and Universal editions, is a COBOL compiler **designed** for the Microsoft (R) .NET framework. The **product generates** Microsoft Intermediate Language (MSIL) that executes under the Common Language Runtime (CLR). The language-neutral, platform- independent environment allows the mixing of COBOL **code** with C# and other languages that support .NET. Fujitsu NetCOBOL for .NET adapts **object** -oriented (OO) COBOL **syntax** and supports .NET-specific extensions. It also integrates with Visual Studio . NET. The system can compile COBOL-85 **code** , allowing users to **employ** existing COBOL assets. The Professional edition of the

program supports the **development** of applications that execute in the .NET environment. The Developer package **includes** the NetCOBOL for Windows Standard edition **product** , which allows users to build Microsoft Windows applications, along with .NET applications. The Universal edition **combines** all the **features** found in the **other** two packages. Fujitsu NetCOBOL for .NET offers full access to .NET framework classes, including XML Web service classes. The system can also debug mixed language applications. It works with Oracle (R), Informix (R), and other databases. The system **includes** project management, reporting, WYSIWYG **design** , text editing, and data file management **features** .

DESCRIPTORS: COBOL Compilers; Debuggers; Enterprise Application
Integration; Program **Development** ; Programming Languages; Text Editors
; Web Services

HARDWARE: Hardware Independent; IBM PC & Compatibles

OPERATING SYSTEM: .NET

PROGRAM LANGUAGES: Active Server Pages; COBOL

TYPE OF PRODUCT: Mini; Micro; Workstation

POTENTIAL USERS: COBOL Developers, Web Services

PRICE: Available upon request

REVISION DATE: 20030228

23/5/5

DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

01029068 DOCUMENT TYPE: Product

PRODUCT NAME: **Eclipse (029068)**

Hayley Enterprise Inc (561215)
1108 Ohio River Blvd
Sewickley, PA 15143 United States
TELEPHONE: (412) 741-6420

RECORD TYPE: Directory

CONTACT: Sales Department

Hayley Enterprise's Eclipse is a knowledge management system for **rule**-based programming. Its **use** of the Rete Algorithm and its widely known Eclipse **syntax** enable it to provide superior functionality to other **rule** systems that require flowcharts to check **rules** . Users can **implement** data- or goal-based reasoning for a variety of languages, including C++ and Java, across any platforms. Eclipse can be used in desktop programs, **embedded** systems, and Web and enterprise server applications. Eclipse eliminates the need to write **code** , **use** flowcharts, or **implement** any **rule** checking procedures. It works by translating **rules** into data- or **event** -driven networks and attaching them to tables or **object** -oriented classes, automatically checking **rules** when any changes are **implemented** . Eclipse **syntax** can easily be used by anyone who has experience with relational database technology or Structured Query Language (SQL). **Additional features include** support for simultaneous goals through automatic subgoaling, automatic memory management, forward and backward chaining, nested representations, **sequences** , and **pattern** matching, and **rule** sets for modular **development** , distribution, and loading. The Eclipse **product** line **includes** CIA Server (TM), Rete++ (TM), and Agent OCX (TM) .

DESCRIPTORS: Artificial Intelligence; Expert Systems; Goal Seeking;
Knowledge Management; Program **Development**

HARDWARE: Apple Macintosh; Hardware Independent; IBM PC & Compatibles;
UNIX

OPERATING SYSTEM: MacOS; ODBC; Open Systems; OS/2; UNIX; Windows; Windows
NT/2000
PROGRAM LANGUAGES: ActiveX; C++; Java
TYPE OF PRODUCT: Mainframe; Mini; Micro; Workstation
POTENTIAL USERS: Developers, Knowledge Management, Event-Driven
Programming
PRICE: Available upon request

REVISION DATE: 20031026

23/5/6
DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

01025381 DOCUMENT TYPE: Product

PRODUCT NAME: 3DLinX (025381)

Global Majic Software Inc (644846)
Bldg 2 #250 620 Discovery Dr
Huntsville, AL 35806 United States
TELEPHONE: (256) 922-0222

RECORD TYPE: Directory

CONTACT: Sales Department

Global Majic Software's 3DLinX enables users to easily **create** 3D applications. 3DLinX builds upon Microsoft's Component **Object** Model (COM) and ActiveX technology to provide a language-independent graphics tool that enables Living Models (TM) to interact with and react to one another and display behaviors and characteristics that follow the laws of physics. 3DLinX is easy to **use**, completely automating the complex functions of 3D application **development** and eliminating the need to write **code**, perform projections, detect collisions, or incorporate complex mathematics. **Additional features** of the **product** include **linking** capabilities, with which users can effectively bind **objects** to one another within a scene, and the ability to expand the **uses** of the **product** through add-ons. Additionally, programmers are able to interact with the scene during **design**, eliminating the need to run applications to test **code**.

DESCRIPTORS: 3D Graphics; Image Processing; Models; Program **Development**

HARDWARE: IBM PC & Compatibles
OPERATING SYSTEM: Windows
PROGRAM LANGUAGES: ActiveX; C++; Visual Basic
TYPE OF PRODUCT: Micro
POTENTIAL USERS: Graphics Developers
PRICE: Available upon request

REVISION DATE: 20010330

23/5/7
DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

00071227 DOCUMENT TYPE: Review

**PRODUCT NAMES: Oracle Cooperative Development Environment (CDE) 2
(440582)**

TITLE: RADical Development : Due This Winter: Oracle CDE2
AUTHOR: Childers, Mark S
SOURCE: PC Magazine, v13 n19 p153(26)(p170) Nov 8, 1994
ISSN: 0888-8509

•
HOMEPAGE: <http://www.pcmag.com>

RECORD TYPE: Review
REVIEW TYPE: Product Analysis
GRADE: Product Analysis, No Rating

Oracle Cooperative Development Environment 2 (CDE 2) has a more powerful feature set, better connectivity, and more ease of use. The product is still a group of bundled tools, including Oracle Forms, Reports, Graphics (all of which include the Object Navigator), Book SQL*Plus, and SQL*Net. CDE2 is one of the first client/server tools with full support for Object Linking and Embedding (OLE) 2.0, for powerful Graph, Form, and BLOB (binary large object) integration. Applications consisting for Forms, Menus, and Libraries, are built with Oracle Forms. Forms are windows with controls, and window controls combine to form Blocks that relate to a basic database table. PL/SQL is the scripting language, and Oracle Reports is a full-functioned report writer that easily manages complicated reports. Chart Genies help with graph creation, and Book creates online help/documentation.

COMPANY NAME: Oracle Corp (010740)
SPECIAL FEATURE: Screen Layouts
DESCRIPTORS: Client/server; Database Management; IDEs; Network Software;
Oracle; Program Development ; SQL
REVISION DATE: 20000430

23/5/8

DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.
(c)2003 Info.Sources Inc. All rts. reserv.

00067142 DOCUMENT TYPE: Review

PRODUCT NAMES: Microsoft Object Linking & Embedding (OLE) (387321

TITLE: Converting VBX controls to OLE custom controls
AUTHOR: Smith, Erick
SOURCE: Microsoft Developer Network News, v3 n4 p1(2) Jul 1994
HOMEPAGE: <http://www.microsoft.com/msdn/news/devne>

RECORD TYPE: Review
REVIEW TYPE: Product Analysis
GRADE: Product Analysis, No Rating

Developers use VBX controls (Visual Basic controls) as a quick way to encapsulate code into reusable components. Microsoft's new control architecture is based on OLE. Developers may wish to convert older VBX controls to OLE custom controls, in order to gain a larger market, 16/32-bit support, and more capabilities. Although Microsoft Access is the only current OLE control container, several other development tools will provide OLE control support in the future. OLE controls are implemented as in-process servers. They use OLE Automation for setting properties and executing methods, and they use type libraries to expose their properties and methods. OLE controls also offer more flexible licensing schemes than VBX. An easy way to convert a VBX to an OLE control is through the OLE Control Development Kit's Control Wizard, which creates a skeleton control based on the existing VBX control.

COMPANY NAME: Microsoft Corp (112127)
SPECIAL FEATURE: Program Listings
DESCRIPTORS: Distributed Objects ; IBM PC & Compatibles; Program
Development ; Visual Basic
REVISION DATE: 20001130

23/5/9

DIALOG(R) File 256:SoftBase:Reviews,Companies&Prods.

(c)2003 Info.Sources Inc. All rts. reserv.

00066570 DOCUMENT TYPE: Review

PRODUCT NAMES: Adobe Persuasion 3.0 (000671)

TITLE: Aldus Persuasion 3.0: A Significant Upgrade

AUTHOR: Bailes, Lenny

SOURCE: Computer Shopper, v14 n7 p502(1) Jul 1994

ISSN: 0886-0556

HOME PAGE: <http://www.computershopper.com>

RECORD TYPE: Review

REVIEW TYPE: Review

GRADE: A

Aldus Persuasion 3.0, two years in the **making**, provides precision editing controls, multimedia **features**, and strong charting options, **making** it a **features** leader on both the Mac and PC. Interface enhancements also **make** the program easier to **use**. Other additions **include** floating palettes and toolbars, a snap-to **ruler** guide, placement- **holder** text prompts, and automatic text kerning. Aldus retained the multiple slide-master system and excellent text outliner. Charting, historically the software's weakest part, is now its strongest. Like its primary competitor, DeltaPoint Graphics's DeltaGraph, Persuasion now offer statistical and mathematical graphing, along with Aldus Table, an integrated OLE (**Object Linking & Embedding**) application. **Other features include** hypertext functions, an Auto-Jump **feature** for opening subsidiary slides, powerful multimedia **features**, and a color-control menu.

PRICE: \$495

COMPANY NAME: Adobe Systems Inc (394173)

SPECIAL FEATURE: Screen Layouts

DESCRIPTORS: Apple Macintosh; Business Graphics; IBM PC & Compatibles;
MacOS; Multimedia; Outliners; Presentations

REVISION DATE: 20010730

Set	Items	Description
S1	1078782	OBJECT? OR CUSTOM(1W)CONTROL? OR SOFTWARE()ROUTINE? OR DLL OR DLLS OR CODE()MODULE? OR VBX OR VBXS OR VISUAL()BASIC()EXT- ENSION? OR READY()MADE() (COMPONENT? OR CONTROL?) OR OCX OR OC- XS OR (ACTIVEX OR ACTIVE()X)()CONTROL?
S2	6607928	GENERAT? OR MAKE? ? OR MAKING OR CREATE? OR CREATING OR PR- ODUC?
S3	1065873	SCRIPT? OR INSTRUCTION? OR RULE? OR SYNTAX OR CODE OR CODES
S4	1220	(MAINTAIN? OR PRESERVE? OR KEEP? ? OR SUSTAIN? OR SUPPORT?-) (3N) (SEPARATELY OR INDEPENDENTLY OR INDIVIDUALLY OR SINGLY)
S5	3110271	WRAP? OR CONTAIN? OR INCLUDE? OR HOLD? OR ENCLOSE?
S6	7022635	ADDITIONAL OR ANOTHER OR SECOND OR 2ND OR DIFFERENT OR MORE OR SECONDARY OR SUPPLEMENT? OR OTHER
S7	7356945	PROPERTY OR PROPERTIES OR FIELD? OR PATTERN? OR QUALITY OR QUALITIES OR ATTRIBUTE? OR FEATURE?
S8	3044033	EVENT? OR HAPPENING? OR INCIDENT? OR OCCURRENCE? OR DEVELO- PMENT? OR EPISODE?
S9	920023	ABOVE OR BEYOND OR EXCEED? OR SURPASS? OR OVER()REACH? OR - OVERREACH?
S10	4967391	INTERNAL OR CONFIGUR? OR DESIGN? OR ARRANGE? OR SET()UP OR SETUP OR PROGRAM?()IN OR SETTING? OR IMBED? OR EMBED? OR FIXED OR ENTRENCH? OR FASTEN? OR INFIX? OR INGRAIN
S11	4703330	UTILIZE OR APPLY? OR APPLIES OR EMPLOY? OR IMPLEMENT? OR U- SE OR USES?
S12	5315516	LINK? OR (PUT OR LAY OR PIECE)()TOGETHER OR JOIN? OR UNITE? OR UNIFY OR UNIFIES OR CONNECT? OR ASSEMBLE? OR COLLECT? OR - COMBINE? OR ASSOCIAT? OR AFFILIAT? OR TIE? OR TRANSACTION?
S13	642352	SEQUENCE? OR QUEUE OR QUEUED OR QUEUING OR QUEUEING OR RAN- K? OR (LOGICAL OR SERIAL)()ORDER OR CONSECUTIVE?
S14	33154	S1 AND S2 AND S3
S15	7230	S5 AND S1 AND (S6 (5N) (S7 OR S8))
S16	159547	(S7 OR S8) AND S10 AND S1
S17	8	S14 AND S4
S18	0	S14 AND S15 AND S16 AND S17 AND S18
S19	270	S14 AND S15 AND S16
S20	0	S19 AND S17
S21	0	S19 AND S18
S22	0	S14 AND S15 AND S17
S23	0	S14 AND S15 AND S18
S24	2958	S1 (5N) S2 (5N) S3
S25	22	S24 AND S15 AND S16
S26	30	S17 OR S25
S27	16	S26 NOT PY>1994
S28	16	S27 NOT PD>19941116
S29	14	RD (unique items)
File	8: Ei Compendex(R) 1970-2003/Dec W1	(c) 2003 Elsevier Eng. Info. Inc.
File	35: Dissertation Abs Online 1861-2003/Nov	(c) 2003 ProQuest Info&Learning
File	202: Info. Sci. & Tech. Abs. 1966-2003/Nov 17	(c) 2003 EBSCO Publishing
File	65: Inside Conferences 1993-2003/Dec W2	(c) 2003 BLDSC all rts. reserv.
File	2: INSPEC 1969-2003/Dec W1	(c) 2003 Institution of Electrical Engineers
File	233: Internet & Personal Comp. Abs. 1981-2003/Jul	(c) 2003, EBSCO Pub.
File	94: JICST-EPlus 1985-2003/Dec W2	(c) 2003 Japan Science and Tech Corp(JST)
File	99: Wilson Appl. Sci & Tech Abs 1983-2003/Nov	(c) 2003 The HW Wilson Co.
File	95: TEME-Technology & Management 1989-2003/Nov W5	(c) 2003 FIZ TECHNIK
File	583: Gale Group Globalbase(TM) 1986-2002/Dec 13	(c) 2002 The Gale Group

29/5/1 (Item 1 from file: 8)
DIALOG(R) File 8:Ei Compendex(R)
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

03935907 E.I. No: EIP94091380565

Title: Method for integrating end-user preferences for design evaluation in rule-based systems

Author: Thurston, D.L.; Crawford, C.A.

Corporate Source: Univ of Illinois, Urbana, IL, USA

Source: Journal of Mechanical Design, Transactions Of the ASME v 116 n 2
Jun 1994. p 522-530

Publication Year: 1994

CODEN: JMDEEC ISSN: 0738-0666

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9410W4

Abstract: Expert systems for design often include provisions for comparison of preliminary design alternatives. Historically, this task has been done on an ad hoc basis (or not at all) due to two difficulties. The first difficulty is design evaluation of multiple attributes. The second is that of taking into account highly subjective end-user preferences. Design experts have developed techniques which have enabled them to deal with these two difficulties; weighted average methods for the former and heuristic 'rules of thumb' which categorize end-users for the latter. Limitations of these techniques are that the accuracy and precision of weighted average methods is inadequate, and that the 'rules of thumb' might be reasonable and valid for most end-users, but not for some others. This paper brings quantitative rigor to the modelling of end-user preferences which is equal to that used in other phases of engineering analysis. We present a technique by which a heuristic rule base derived from technical experts can be analyzed and modified to integrate quantitative assessment of end-users' subjective preferences. The operations research tool of multiattribute utility analysis is integrated with artificial intelligence techniques to facilitate preliminary evaluation of design alternatives of multiple attributes specific to individual users. The steps of the methodology are: develop the heuristic rule base, analyze the rule base to separate subjective from objective rules, add a subjective multiattribute utility assessment module, add an uncertainty assessment module, make objective rules explicit, and express performance attributes in terms of design decision variables. The key step is making the distinction between subjective and objective aspects of rules, and replacing the former with utility analysis. These steps are illustrated through an expert system for materials selection for a sailboat mast. Results indicate improved expert system performance for both 'typical' and 'atypical' end-users. (Author abstract) 56 Refs.

Descriptors: Knowledge based systems; Machine design; Heuristic methods; Operations research; Masts (boat); Computer aided design; Automation; Performance; Decision theory; Expert systems

Identifiers: End user preferences; Design evaluation; Heuristic rule base; Rules of thumb

Classification Codes:

723.4.1 (Expert Systems)

723.4 (Artificial Intelligence); 921.6 (Numerical Methods); 912.3 (Operations Research); 671.1 (Ship Design); 723.5 (Computer Applications)

723 (Computer Software); 601 (Mechanical Design); 921 (Applied Mathematics); 912 (Industrial Engineering & Management); 671 (Naval Architecture)

72 (COMPUTERS & DATA PROCESSING); 60 (MECHANICAL ENGINEERING); 92 (ENGINEERING MATHEMATICS); 91 (ENGINEERING MANAGEMENT); 67 (MARINE ENGINEERING)

29/5/2 (Item 2 from file: 8)
DIALOG(R) File 8:Ei Compendex(R)
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

03932016 E.I. No: EIP94081374626

Title: Development of generic field classes for finite element and finite difference problems

Author: Verner, Diane A.; Heileman, Gregory L.; Budge, Kent G.; Robinson, Allen C.

Corporate Source: Univ of New Mexico, Albuquerque, NM, USA

Source: Scientific Programming v 2 n 4 Winter 1993. p 227-234

Publication Year: 1993

CODEN: SCIP EV ISSN: 1058-9244

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9410W3

Abstract: This article considers the development of a reusable **object**-oriented array library, as well as the use of this library in the construction of finite difference and finite element **codes**. The classes in this array library are also generic enough to be used to construct other classes specific to finite difference and finite element methods. We demonstrate the usefulness of this library by inserting it into two existing **object**-oriented scientific **codes** developed at Sandia National Laboratories. One of these **codes** is based on finite difference methods, whereas the other is based on finite element methods. Previously, these **codes** were **separately maintained** across a variety of sequential and parallel computing platforms. The use of **object**-oriented programming allows both **codes** to **make** use of common base classes. This offers a number of advantages related to optimization and portability. Optimization efforts, particularly important in large scientific **codes**, can be focused on a single library. Furthermore, by encapsulating machine dependencies within this library, the optimization of both **codes** on different architectures will only involve modification to a single library. (Author abstract) 11 Refs.

Descriptors: Equivalence classes; **Object** oriented programming; Finite element method; Finite difference method; **Codes** (symbols); Parallel processing systems; Sequential machines; Optimization; Computer software portability; Computer architecture

Identifiers: Generic field classes; **Object** oriented array library; Scientific **codes**; Sequential computing platforms; Parallel computing platforms

Classification Codes:

721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.1 (Computer Programming); 921.6 (Numerical Methods); 722.4 (Digital Computers & Systems); 921.5 (Optimization Techniques); 723.2 (Data Processing)
721 (Computer Circuits & Logic Elements); 723 (Computer Software); 921 (Applied Mathematics); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

29/5/3 (Item 3 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

03765378 E.I. No: EIP93071030674

Title: Exploring class reuse and integration for hybrid simulation in C plus plus

Author: Hitt, Teresa L.; Reilly, Kevin D.

Corporate Source: Jefferson State Community Coll, Birmingham, AL, USA

Conference Title: Proceedings of the 3rd Workshop on Neural Networks: Academic/Industrial/NASA/Defense

Conference Location: Alabama, AL, USA Conference Date: 19920210-19920212

Sponsor: Auburn University Space Power Institute; Center for Commercial Development of Space Power and Advanced Electronics; NASA Headquarters

E.I. Conference No.: 18706

Source: Proceedings of SPIE - The International Society for Optical Engineering v 1721 1993. Publ by Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, USA. p 449-454

Publication Year: 1993
CODEN: PSISDG ISSN: 0277-786X ISBN: 1-56555-007-2
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical); A;
(Applications)
Journal Announcement: 9402W1

Abstract: Currently, a relatively large number of simulation programs are written in special purpose code, when general-purpose simulation languages should suffice. One reason for this is that, in the minds of most workers, these languages fail to support components they require to adequately represent the systems under study. Hybrid models in particular, which may contain a symbolic, and/or artificial neural network (ANN) ones) present even a greater problem. The hybrid-model developer who would like to exploit existing code faces the challenge of reconciling code that is available with that he-is developing. In our work, most interest is in merging existing code written in the **object** -oriented language C plus plus - with its promise of reusability and **other** good **qualities** - with **code** we are **generating** . We have considered several **object** -oriented and C plus plus simulation systems (or portions thereof) in search of reusable code and **designs** . This paper discusses both positive and negative aspects of this code (and its **design**) with respect to merging it with our own. It also outlines the tactics we have employed to facilitate reusing special purpose code and demonstrates the use of templates in C plus plus for building reusable simulation and simulation support code. (Author abstract) 20 Refs.

Descriptors: *C (programming language); Computer simulation; Computer programming; General purpose computers; Neural networks

Identifiers: Hybrid simulation; General-purpose simulation languages; Reusable simulation

Classification Codes:

723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 723.5 (Computer Applications)
723 (Computer Software); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING)

29/5/4 (Item 4 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

02082337 E.I. Monthly No: EIM8604-019876

Title: **CAPABILITY ARCHITECTURE FOR ADA.**

Author: Biswas, Prasenjit

Corporate Source: Southern Methodist Univ, Dallas, TX, USA

Conference Title: IEEE Computer Society 1984 Conference on Ada Applications and Environments.

Conference Location: St. Paul, MN, USA Conference Date: 19841015

Sponsor: IEEE Computer Soc, Computer Languages Technical Committee, Los Alamitos, CA, USA.

E.I. Conference No.: 07731

Source: Publ by IEEE, New York, NY, USA Available from IEEE Service Cent (Cat n 84CH2083-4), Piscataway, NJ, USA p 23-32

Publication Year: 1984

ISBN: 0-8186-0590-1

Language: English

Document Type: PA; (Conference Paper)

Journal Announcement: 8604

Abstract: A report is presented of the **design** of a capability-based stack processor architecture that reduces the semantic gap between programs written in Ada and the **object code produced** . One of the important **features** in Ada is its facility for data abstraction. Moreover, a highly desirable characteristic of multiuser computing environment is the availability of support for efficient execution of a process in a number of small protection domains. Both of these **features** are simultaneously supported in the proposed architecture using tagged capabilities and through the definition of hardware/firmware recognized **objects** called packets. **Other** important **features** in the architecture **include** an

efficient support for variable addressing mechanism, support for parameter passing mechanism in Ada, and architectural **features** to support run-time constraint checking and representation of discriminant records/dynamic arrays. 26 refs.

Descriptors: *COMPUTER OPERATING SYSTEMS--*Program Processors; COMPUTER PROGRAMMING LANGUAGES--Ada; COMPUTER ARCHITECTURE

Identifiers: COMPUTER LANGUAGE PROCESSORS

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

29/5/5 (Item 5 from file: 8)

DIALOG(R) File 8:EI Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

00912228 E.I. Monthly No: EI8004027867 E.I. Yearly No: EI80015888

Title: **SOFTWARE ENGINEERING -- A NEW APPROACH.**

Author: Pearson, David

Corporate Source: Bell North Res

Source: Telesis v 6 n 5 Oct 1979 p 23-27

Publication Year: 1979

CODEN: TLSSAO ISSN: 0040-2710

Language: ENGLISH

Journal Announcement: 8004

Abstract: The **objective** of a new integrated software engineering system, Ises, is to control the total life cycle **development** of a software product. At the heart of Ises is the life cycle data base which **contains** information about the system's requirements, its **design**, the computer program's source code and **object** code as well as trouble reports from the **field**. This data base 'drives' all the major tools involved in the **development** of the software product. There are several interfaces between the user and Ises: one allows the **designer** to capture a **design** in either textual or graphical form; another is capable of evaluating the accuracy and **quality** of the **product** at each stage in its **development**; and yet **another** constructs source **code modules** in the appropriate programming language. The 'prompts' in the **design** capture system and its built-in sequencing and standards rules make it possible for inexperienced **designers** to produce good **designs**. The semi-automatic generation of code reduces the number of coding errors and increases productivity.

Descriptors: COMPUTER PROGRAMMING; COMPUTER PROGRAMS-- **Design**

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

29/5/6 (Item 6 from file: 8)

DIALOG(R) File 8:EI Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

00453518 E.I. Monthly No: EI7505029822 E.I. Yearly No: EI75018466

Title: **GUIDE FOR DEVELOPING PERFORMANCE-STRUCTURE ORIENTED COMPUTER-ADMINISTERED INSTRUCTION: THEORY AND RESEARCH.**

Author: Blaiwes, Arthur S.; Rigney, J. W.

Corporate Source: Nav Train Equip Cent, Orlando, Fla

Source: Hum Factors Soc, Annu Meet, 18th, Proc, Huntsville, Ala, Oct 15-17 1974 p 441-451. Publ by Hum Factors Soc, Santa Monica, Calif, 1974

Publication Year: 1974

Language: ENGLISH

Journal Announcement: 7505

Abstract: This work has three main **objectives** with respect to computer-administered **instruction** (CAI), viz. , to **make** CAI: (1) easier to develop, (2) higher in **quality**, and (3) **more** widely appreciated and accepted. The approach taken toward these goals **included** two complementary kinds of efforts. One effort was directed toward **designing** guidelines and models of CAI which can assist training program developers in their efforts to implement CAI. Another effort was applied to the

construction and **field** evaluation of a trainer which was based on descriptions and principles of CAI as **contained** in the guidelines and models. These efforts and some of their results are discussed here. 6 refs.

Descriptors: *DATA PROCESSING--*Educational Applications

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

29/5/7 (Item 1 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2003 ProQuest Info&Learning. All rts. reserv.

01363103 ORDER NO: AAD94-18186

AN EXECUTABLE SPECIFICATIONS APPROACH TO COMPONENT-ORIENTED SOFTWARE DEVELOPMENT

Author: KEEBLER, FREDERICK MATT

Degree: PH.D.

Year: 1993

Corporate Source/Institution: WAYNE STATE UNIVERSITY (0254)

Adviser: S. P. RANA

Source: VOLUME 55/02-B OF DISSERTATION ABSTRACTS INTERNATIONAL.

PAGE 493. 220 PAGES

Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

Software **development** activity is facing **more** acute challenges now than ever in the past. Strong competition and highly volatile requirements call for shorter **development** times and an order of magnitude of improvements in software productivity and **quality**. Recently there has been an upsurge of interest in software reuse and **object**-oriented technology. Each of the existing reuse techniques has its merits and demerits. Component-oriented software **development** approaches attempt to use a combination of techniques to reap their benefits while alleviating their disadvantages and thus **holds** the promise for a solution to the software crisis.

In this dissertation we presented a component-oriented software **development** methodology, called ESDM, which emphasizes the compositional approach to software **development**. To our knowledge, ESDM is the most comprehensive approach that exploits the strengths of existing technologies such as **object**-oriented **development**, executable specifications, reusable schemas, and automatic **code generation**. The generality of ESDM allows for interoperability among software components developed using **different** paradigms.

A unique **feature** of the ESDM methodology is the ability to proceed with partial and even inconsistent statement of requirements. Unlike traditional approaches, it does not have any sharp boundaries between requirement specification, **design**, and implementation stages. ESDM embodies the so-called "concurrent engineering" philosophy, where all phases of the software **development** proceed simultaneously in unison and active collaboration with the end user. This active collaboration with the end user leads to a reduction in the communication gap. The net impact is a greater acceptance by the user and the drastic reduction in complexity of the costly test-revise-fix cycle. It is this reduction in latency time of an iteration that allows ESDM to proceed and revise requirements with ease at any stage, thus reducing the **development** and maintenance time.

29/5/8 (Item 2 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2003 ProQuest Info&Learning. All rts. reserv.

849774 ORDER NO: AAD84-16588

LANGUAGE CONSTRUCTS FOR THE MANAGEMENT OF DISTRIBUTED COMPUTATIONS

Author: HELIOTIS, JAMES ELIAS

Degree: PH.D.

Year: 1984

Corporate Source/Institution: THE UNIVERSITY OF ROCHESTER (0188)
Source: VOLUME 45/04-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 1236. 213 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

This thesis involves methods of organization of distributed computations in a heterogeneous network. **More** specifically, it is the **development** of enhancements to high-level languages to make them suitable for use in such an environment, including constructs for organization of **objects**, failure notification, and job/service recovery.

We first propose a new concept called an activity. It is **designed** to group **objects** together in any **arrangement** that a programmer sees fit, in order to increase the amount of control the programmer has over his software, and improve its reliability.

We assume an **object**-oriented programming system, where all program code is **contained** within **objects**, and where some **objects** are actually implementors of an **object** type. To add activities to a programming system, some new items need to be added. First, we need a network-wide activity coordinator which maintains the registration lists, implementor, manager, and ACM ports, and the activity tree structure. The **object** implementors must be modified to accept communications from the activity coordinator (**object** and activity commands). Finally, we allow a new distinction for **objects**, that of being an(other) **object**'s manager.

We then **designed** language constructs specifically for an activity system to help make it easier for a user to take full advantage of that system. First, the distributed programming paradigm of PLITS (Fel79) is chosen. In PLITS, **objects** called modules are **created** which **contain** both **code** and process state. Second, the language syntax of Mesa (Mit79) is chosen because of the constructs it already **contains** to handle intra-process exceptions. These are expanded to allow general inter-process (or inter-**object**) notifications. Aside from the PLITS constructs that had to be inserted into the Mesa syntax, the main additions were new statements/expressions to invoke activity and **object** commands, notice catch phrases for receiving those commands and emergency notices, and increased use of **object** IDs and activity tags in tagged variables and iterative loops.

29/5/9 (Item 3 from file: 35)
DIALOG(R) File 35:Dissertation Abs Online
(c) 2003 ProQuest Info&Learning. All rts. reserv.

741656 ORDER NO: AAD81-07413

THE LANGUAGES OF THE FOREIGN LANGUAGE CLASSROOM: A STUDY OF TEACHER USE OF THE NATIVE AND TARGET LANGUAGES FOR LINGUISTIC AND COMMUNICATIVE FUNCTIONS

Author: WING, BARBARA HAMMANN

Degree: PH.D.

Year: 1980

Corporate Source/Institution: THE OHIO STATE UNIVERSITY (0168)

Source: VOLUME 41/10-A OF DISSERTATION ABSTRACTS INTERNATIONAL.

PAGE 4388. 312 PAGES

Descriptors: LANGUAGE, LINGUISTICS

Descriptor Codes: 0290

Language use in the foreign language classroom is a critical variable of significant importance and complexity. The presence of two languages, native and target, and the dual **instructional objectives** of linguistic and communicative competency **create** a unique educational **setting**. Systematic observation of the language use variable in this **setting** can benefit researchers, methodologists, and teachers.

This ex post facto study validated the Linguistic/Communicative-Functions Analysis System (L/C-FAS), an observational system developed by the researcher for describing teacher verbal behavior in the foreign language classroom. It also investigated the possibility of relationships between **patterns** of teacher native/target

Editor(s): Karagiannis, D.
 Publisher: Springer-Verlag, Berlin, Germany
 Publication Date: 1994 Country of Publication: West Germany xvii+807
 pp.
 ISBN: 3 540 58435 8
 Conference Title: Proceedings of DEXA 94
 Conference Date: 7-9 Sept. 1994 Conference Location: Athens, Greece
 Language: English Document Type: Conference Paper (PA)
 Treatment: Applications (A); Practical (P)
 Abstract: The lack of a well- **designed** class hierarchy **holds** back the **development** of an **object** -oriented database (OODB) for genome information. This paper describes the **design** of a standard class hierarchy for genome information with **object** -oriented technology and the integration of genome information stored in different database files into an OODB. First, we define the **rules** for finding **objects** and **creating** classes for genome information, which can be put to use building new classes in **other** **fields** of biological sciences. **Second** , we implement more than 80 classes for genome information. Finally, we demonstrate the importance of well- **designed** classes in the **development** of an OODB and that of the good graphical user interface. (7 Refs)
 Subfile: C
 Descriptors: biology computing; genetics; graphical user interfaces; **object** -oriented databases; systems analysis
 Identifiers: human genome databases; **object** -oriented database; database **design** ; well- **designed** class hierarchy; genome information; standard class hierarchy; database files; biological sciences; well- **designed** classes; graphical user interface
 Class Codes: C7330 (Biology and medical computing); C6110 (Systems analysis and programming); C6160J (Object-oriented databases); C6180G (Graphical user interfaces)
 Copyright 1995, IEE

29/5/11 (Item 2 from file: 2)
 DIALOG(R)File 2:INSPEC
 (c) 2003 Institution of Electrical Engineers. All rts. reserv.

04335763 INSPEC Abstract Number: C9303-7320-090
Title: Construction of large-scale simulation codes using ALPAL:
 generation of a hydrodynamics code module
 Author(s): Cook, G.O., Jr.; Painter, J.F.
 Author Affiliation: Lawrence Livermore Nat. Lab., CA, USA
 Conference Title: Artificial Intelligence, Expert Systems and Symbolic Computing. Selected and Revised Papers from the IMACS 13th World Congress
 p.82-8
 Editor(s): Houstis, E.N.; Rice, J.R.
 Publisher: North-Holland, Amsterdam, Netherlands
 Publication Date: 1992 Country of Publication: Netherlands xii+458
 pp.
 ISBN: 0 444 89703 8
 Conference Date: 22-26 July 1991 Conference Location: Dublin, Ireland
 Language: English Document Type: Conference Paper (PA)
 Treatment: Applications (A); Practical (P)
 Abstract: A Livermore Physics Applications Language (ALPAL) is a tool that automatically generates code to solve nonlinear integro-differential equations, given a very high-level specification of the equations to be solved and the numerical methods to be used. ALPAL is **designed** to handle the sort of complicated mathematical models used in very large scientific and engineering simulation codes. **Other features** of ALPAL **include** an interactive graphical front end, the ability to symbolically compute exact Jacobian for implicit methods, and a high degree of optimization in generated code. The authors show how ALPAL is used in practice by using it to construct a simple one-dimensional hydrodynamics code. (3 Refs)
 Subfile: C
 Descriptors: codes; graphical user interfaces; hydrodynamics; integro-differential equations; physics computing; software tools
 Identifiers: large-scale simulation codes; ALPAL; hydrodynamics **code**

module ; Livermore Physics Applications Language; tool; nonlinear
integro-differential equations; mathematical models; interactive graphical
front end; Jacobian

Class Codes: C7320 (Physics and Chemistry); C6115 (Programming support)
; C4180 (Integral equations); C4170 (Differential equations); C6180G (
Graphical user interfaces)

29/5/12 (Item 1 from file: 233)

DIALOG(R)File 233:Internet & Personal Comp. Abs.

(c) 2003, EBSCO Pub. All rts. reserv.

00227807 90JR10-005

Transaction shells: A new approach to courseware authoring

Zhongmin Li; Merrill, M David

Journal of Research on Computing in Education , October 1, 1990 , v23 n1
p72-86, 15 Page(s)

ISSN: 0888-6504

Languages: English

Document Type: Feature Articles and News

Geographic Location: United States

Describes a transaction shell approach to the development of
educational courseware. Says a transaction shell is an integrated
instructional environment **designed** for a set of clearly defined
instructional objectives . Also says transaction shells **make**
courseware **development** **more** efficient, and that they **include**
strategies more effective than those possible with authoring systems or
languages. **Includes** two diagrams, two tables, one screen display. (tbc)

Descriptors: Courseware; Programming **Design** ; Educational Computing;
Research; Product **Development**

29/5/13 (Item 2 from file: 233)

DIALOG(R)File 233:Internet & Personal Comp. Abs.

(c) 2003, EBSCO Pub. All rts. reserv.

00196482 89CP07-010

Complete C: a shareware OOP C

Parker, Tom

Computer Language , July 1, 1989 , v6 n7 p117-118, 2 Pages

ISSN: 0749-2839

Languages: English

Document Type: Software Review

Grade (of Product Reviewed): B

Geographic Location: United States

Presents a favorable review of Complete C (\$449), an **object** -oriented C
programming system from Complete Computer Corp. of New York, NY (212). Says
it provides OOP programming in standard C; it is available as a shareware
version and a commercial release, which are identical except for supported
memory versions. Says it **generates** source **code** which must be compiled
separately ; **supported** by a manual and bulletin board and telephone
support; provides two modes: development which provides extensive debugging
and tracing capabilities and release which provides run-time speed and
code -size optimization. It is supported by utilities which **make**
programming easier. (v1)

Descriptors: C Programming Language; **Object** -oriented Languages;
Software Review

Identifiers: Complete C; Complete Computer

29/5/14 (Item 1 from file: 95)

DIALOG(R)File 95:TEME-Technology & Management

(c) 2003 FIZ TECHNIK. All rts. reserv.

00715191 E93104476080

Exploring class reuse and integration for hybrid simulation in C++

(Entdeckung der Klassenwiederverwendung und Integration fuer die hybride

Simulation in C++)

Hitt, TL; Reilly, KD

Jefferson State Community College, Birmingham, USA; Univ. Alabama,
Birmingham, USA

WNN 92, 3rd Workshop on Neural Networks: Academic/Industrial/NASA/Defense,
Auburn, AL, USA, 10-12 Feb., 1992; South Shore Harbour, TX, USA, 4-6 Nov
1992

Document type: Conference paper Language: English

Record type: Abstract

ABSTRACT:

Currently, a relatively large number of simulation programs are written in special purpose code, when general-purpose simulation languages should suffice. One reason for this is that, in the minds of most workers, these languages fail to support components they require to adequately represent the systems they require to adequately represent the systems under study. Hybrid models in particular, which may **contain** a variety of components (e.g., discrete, continuous, symbolic, and/or artificial neural network (ANN) ones) present even a greater problem. The hybrid-model developer who would like to exploit existing code faces the challenge of reconciling code that is available with that he is developing. In this work, most interest is in merging existing code written in the **object** -oriented language C++ - with its promise of reusability and **other** good **qualities** - with **code** we are **generating** . Several **object** -oriented and C++ simulation systems are considered (or portions thereof) in search of reusable code and **designs** . This paper discusses both positive and negative aspects of this code (and its **design**) with respect to merging it with our own. It also outlines the tactics we have employed to facilitate reusing special purpose code and demonstrates the use of templates in C++ for building reusable simulation and simulation support code.

DESCRIPTORS: HYBRID SIMULATION; SYSTEM SIMULATION; MODEL SIMULATION;
SIMULATION LANGUAGES; **OBJECT** ORIENTED PROGRAMMING; ARTIFICIAL NEURAL
NETWORKS; ARTIFICIAL INTELLIGENCE; APPLICATION SOFTWARE; C PLUS PLUS--
PROGRAMMING LANGUAGE; PROGRAM REUSABILITY

IDENTIFIERS: hybride Simulation; C++; Wiederverwendung

Set	Items	Description
S1	563683	OBJECT? OR CUSTOM(1W)CONTROL? OR SOFTWARE()ROUTINE? OR DLL OR DLLS OR CODE()MODULE? OR VBX OR VBXS OR VISUAL()BASIC()EXTENSION? OR READY()MADE() (COMPONENT? OR CONTROL?) OR OCX OR OCXS OR (ACTIVEX OR ACTIVE()X)()CONTROL?
S2	5477547	GENERAT? OR MAKE? ? OR MAKING OR CREATE? OR CREATING OR PRODUCE?
S3	1163157	SCRIPT? OR INSTRUCTION? OR RULE? OR SYNTAX OR CODE OR CODES
S4	2724	(MAINTAIN? OR PRESERVE? OR KEEP? ? OR SUSTAIN? OR SUPPORT?-(3N) (SEPARATELY OR INDEPENDENTLY OR INDIVIDUALLY OR SINGLY)
S5	3210709	WRAP? OR CONTAIN? OR INCLUDE? OR HOLD? OR ENCLOSE?
S6	5623657	ADDITIONAL OR ANOTHER OR SECOND OR 2ND OR DIFFERENT OR MORE OR SECONDARY OR SUPPLEMENT? OR OTHER
S7	2428573	PROPERTY OR PROPERTIES OR FIELD? OR PATTERN? OR QUALITY OR QUALITIES OR ATTRIBUTE? OR FEATURE?
S8	2457829	EVENT? OR HAPPENING? OR INCIDENT? OR OCCURRENCE? OR DEVELOPMENT? OR EPISODE?
S9	906091	ABOVE OR BEYOND OR EXCEED? OR SURPASS? OR OVER()REACH? OR -OVERREACH?
S10	2766882	INTERNAL OR CONFIGUR? OR DESIGN? OR ARRANGE? OR SET()UP OR SETUP OR PROGRAM?()IN OR SETTING? OR IMBED? OR EMBED? OR FIXED OR ENTRENCH? OR FASTEN? OR INFIX? OR INGRAIN
S11	3465569	UTILIZE OR APPLY? OR APPLIES OR EMPLOY? OR IMPLEMENT? OR USE OR USES?
S12	5236095	LINK? OR (PUT OR LAY OR PIECE)()TOGETHER OR JOIN? OR UNITE? OR UNIFY OR UNIFIES OR CONNECT? OR ASSEMBLE? OR COLLECT? OR -COMBINE? OR ASSOCIAT? OR AFFILIAT? OR TIE? OR TRANSACTION?
S13	393292	SEQUENCE? OR QUEUE OR QUEUED OR QUEUING OR QUEUEING OR RANK? OR (LOGICAL OR SERIAL)()ORDER OR CONSECUTIVE?
S14	32964	S1 (S) S2 (S) S3
S15	7636	S5 (S) S1 (S) (S6 (5N) (S7 OR S8))
S16	53213	(S7 OR S8) (S) S10 (S) S1
S17	139872	S11 (S) (S6 (5N) (S7 OR S8))
S18	157256	(S12 OR S13) (S) S1
S19	10	S14 (S) S4
S20	466	S14 (S) S15 (S) S16 (S) S17 (S) S18
S21	6880	S1 (5N) S2 (5N) S3
S22	47	S21 (S) S14 (S) S16 (S) S17 (S) S18
S23	24	S22 NOT PY>1994
S24	23	S23 NOT PD>19941116
S25	20	RD (unique items)

File 647:CMP Computer Fulltext 1988-2003/Dec W2

(c) 2003 CMP Media, LLC

File 275:Gale Group Computer DB(TM) 1983-2003/Dec 17

(c) 2003 The Gale Group

File 674:Computer News Fulltext 1989-2003/Dec W1

(c) 2003 IDG Communications

File 696:DIALOG Telecom. Newsletters 1995-2003/Dec 16

(c) 2003 The Dialog Corp.

File 15:ABI/Inform(R) 1971-2003/Dec 17

(c) 2003 ProQuest Info&Learning

File 636:Gale Group Newsletter DB(TM) 1987-2003/Dec 17

(c) 2003 The Gale Group

25/5,K/1 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext
(c) 2003 CMP Media, LLC. All rts. reserv.

01034500 CMP ACCESSION NUMBER: CRN19941107S0023
TALIGENT UNVEILS OBJECT ENVIRONMENT (BETWEEN THE LINES)
HEATHER CLANCY
COMPUTER RESELLER NEWS, 1994, n 603, PG30
PUBLICATION DATE: 941107
JOURNAL CODE: CRN LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: NEWS
WORD COUNT: 218
TEXT:

SAN MATEO, CALIF.-Taligent Inc. is expected to roll out at Comdex/
Fall its Taligent Application Environment, known as TalAE.

... operating systems, such as OS/2, AIX , HPUX and Apple's forthcoming
PowerOpen, letting developers or VARs **assemble** a single set of **code**
for multiple platforms. It also **employs** **object**-oriented **design**
models, allowing developers to **create** **code** **more** quickly than with
existing **development** methods.

A half-dozen TalAE solutions are to be shown at Comdex, including
whiteboard and collaborative-computing...

25/5,K/2 (Item 2 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext
(c) 2003 CMP Media, LLC. All rts. reserv.

01022355 CMP ACCESSION NUMBER: NWC19940601S2089
CASE Products Swing Toward Client/Server
Alan Radding
NETWORK COMPUTING, 1994, n 507 , 73
PUBLICATION DATE: 940601
JOURNAL CODE: NWC LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Client/Server Databases
TEXT:

Developers who thought they were leaving CASE tools behind when they
jumped into the iterative prototyping, rapid application **development**
world of client/server computing are discovering that, like it or not,
Computer-Aided Software Engineering (CASE) tools are tagging right along.
Client/server versions of leading host-based CASE tools are shipping
already, and more CASE **products** for client/server computing are in the
pipeline. From the developers' standpoint, CASE for client/server
computing draws decidedly mixed reviews. Where host-based CASE tools
forced developers to follow detailed, rigid methodologies, the graphical
nature of client/server computing encourages fast, seat-of-the-pants
iterative prototyping. Where host-based CASE tools tried to capture and
generate the entire application at once in monolithic, procedural **code** ,
client/server computing demands small, highly modular and partitioned
pieces of **code** to **produce** **event**-driven applications. Nevertheless,
fast, iterative prototyping and seat-of-the-pants techniques can take
developers only so far. "Client/server applications are more complicated
than conventional applications," says Judith Hurwitz, president of Hurwitz
Consulting, Watertown, Mass., and publisher of the Client/Server Tool
Watch newsletter. " **More** things are **happening** . There are **more** and
different components. Developers really need tools to help them **create**
a structure." Those tools are client/server CASE **products** . Effective
client/server CASE **products** let developers capture the application's
business requirements easily, define business **rules** and **design** the
application quickly, either by drawing the components and relationships
between them using graphical tools or by specifying them using an
English-like language and point-and-click tools. The CASE tool also must
provide for bidirectional communication between the **design** and build
stages, so changes made in one are reflected automatically in the other.

The CASE Difference Client/server CASE tools differ from conventional CASE tools in a number of ways. First, the **design objective** of conventional CASE tools is to identify, predict and capture all the requirements of the finished application from the outset. Client/server CASE tools, by contrast, are **designed** to facilitate change, primarily through the repository and two-way information flow. Client/server CASE tools also differ from conventional CASE tools in the modularity of the **design** and **code** they **produce**. Conventional CASE tools **generate** monolithic, procedural applications that run on the host. Client/server tools **generate** modular **code** and partition the application among client, server and middleware components. Finally, client/server CASE tools differ from their conventional counterparts in terms of cross-platform support. Today's client/server environment is highly heterogeneous at every level. The CASE tools and the applications they **generate** must be able to run on a wide range of client and server platforms, and access a variety of relational and nonrelational databases. **Making** a CASE for C/S The earliest client/server tools were disappointing. "In a lot of cases, the vendors simply put a graphical front end on their conventional CASE tool," says Hurwitz. Ed Acly, senior analyst at International Data Corp., Framingham, Mass., says, "At first, CASE for client/server meant lower CASE." Lower CASE tools automate the **code generation** and construction stage of **development** but lack the business modeling and **design** component, referred to as upper CASE. A second **generation** of client/server tools, however, addresses the full range of CASE requirements. According to Hurwitz, the key components of client/server CASE **products** include: /\ Repository or knowledge base. A central source where business **rules** are articulated and the application components are managed to facilitate change and, where possible, reuse. /\ Version control and **configuration** management. A method for tracking different versions of the application and individual components, and coordinating the work of multiple developers. /\ Automatic **code generation** and partitioning. The capability to **generate** CASE **designs** and repository **objects** as working client, server and middleware **code**. /\ Two-way communication flow. Between the **design** components and build components, usually via the repository. /\ Cross-platform support, or middleware. Handles multiple clients, servers and databases across a variety of networks. /\ Published APIs. Allows **design** elements, definitions or application **objects** created with another tool to be incorporated into the new application. These components comprise a framework for evaluating client/server CASE tools. No **product** yet delivers all the key components well, but the following deliver enough to be considered effective client/server CASE tools. Andersen Foundation for Cooperative Processing Andersen Consulting's Foundation for Cooperative Processing (FCP) is the client/server version of Andersen's conventional Foundation CASE **product**. Like the host **product**, FCP provides both upper and lower CASE capabilities and is intended for large-scale **development**. FCP uses Andersen's Method/1 methodology, a big, complex methodology, like traditional CASE, that walks developers through initial business information planning, **design** strategy, data modeling, project definition and more. You don't have to **use** the methodology at all, says Byron Miller, FCP developer, but he recommends that you **use** a subset of the methodology, at least. In terms of the above framework, FCP provides most of the pieces to some degree. The Foundation Repository maintains the voluminous information **generated** by Foundation's data modeling, Entity-Relationship modeling, and other modeling and diagramming **design** components. The Repository stores the information as **objects** that can be reused in other FCP applications. The information flow, however, is one-way only, from **design** to the Repository to the build component. Although the data in the Repository can be read using standard SQL, FCP's Repository is not open. Only FCP modules **make use** of the information as it is stored in the Repository, and information cannot be imported from or exported to other tools. FCP provides a messaging architecture and Applications Programming Interface (API) to support cross-platform **development**. Foundation, however, supports only a relatively small number of clients, servers and databases (see chart on page 88). FCP also lacks **configuration** and versioning control, although Miller says it will be added in the future. Texas Instruments IEF for Client/Server IEF for

Client/Server is an extension of Texas Instruments' host IEF tools. A more recent entry, it delivers some of the **features** lacking in FCP, such as version control and **configuration** management for workgroups. IEF for Client/Server also partitions applications between client and server automatically. Like FCP, it is methodology-based and driven by **design** models stored in a repository called the Encyclopedia. **Designs** are then **generated** as application **code**. Subsequent changes are made to the **design**, and the system regenerates new **code**. "The auto **code** **generation** produces 100 percent syntactically correct **code**, but we still have to correct **design** logic errors," says Gerry Wethington, director of information systems for the Missouri State Highway Patrol, Jefferson City, Mo. All changes are made using the IEF **design** tools. The tool does not support two-way information flow between **design** and build phases. IEF supports cross-platform **development**. An alliance with PeerLogic promises to enhance IEF middleware **connectivity** capabilities. TI also promises to extend platform support significantly and provide an open API to enable other tools to leverage information in the Encyclopedia. LBMS System Engineer LBMS' System Engineer differs markedly from the other two tools. Based on LBMS' conventional upper CASE tools, it provides **design** and analysis capabilities for popular 4GL client/server **development** tools, starting with PowerBuilder and soon SQLWindows and Uniface. System Engineer also differs by providing two-way communication between its repository and the **design** and build components. For example, a developer **uses** the System Engineer **design** tool to **design** an application that is then **generated** automatically for PowerBuilder. The developer can **make** changes in either PowerBuilder or System Engineer, and the changes are captured by and updated in the repository, keeping the **design** and the application in synch. "The bidirectional communication is very important to us and saves a lot of work. People here don't like to go back to the **design** tool! and **make** changes," says Philip Fasone, senior manager of **quality** assurance at Holiday Inn Worldwide Information Technologies, Atlanta. The United Nations Children's Fund (UNICEF), New York, began developing a major application with PowerBuilder before getting the bidirectional **link** to System Engineer. "Even though 25 percent of the application was already done, we were able to back the information into the System Engineer repository," says Frank Buchanan, data administration and planning officer. The application **design** was then completed in System Engineer, which **generates** the basic PowerBuilder libraries automatically. The repository is the heart of System Engineer. Although built on the Gupta SQL database, the repository is not an open platform but can be used only with the **development** tools with which it is tightly integrated, such as PowerBuilder. The repository imposes some version control on the process by managing access to **objects** and identifying groups of **objects** that together form a complete, deliverable application. System Engineer's cross-platform support is limited to the platforms and databases supported by PowerBuilder and SQLWindows. More To Come Developers can expect more client/server CASE tool choices in the future. Although the current crop of client/server CASE **products** addresses most of the basic requirements of client/server computing, there is still plenty of room for improvement, particularly in the areas of two-way communication, an open repository and cross-platform support. Of all the new CASE **products** for client/server, the most exciting twist is the way in which LBMS' System Engineer brings integrated CASE to a popular client/server application **development** tool like PowerBuilder. Leverage what you have and what you know into full-blown CASE, or ease into client/server with an application **development** tool and add CASE incrementally later on. Either approach **makes** a lot of sense. Alan Radding is a freelance writer, based in Newton, Mass., who specializes in application **development** and networking. He can be reached on CompuServe at 72320,2763.

TEXT:

... who thought they were leaving CASE tools behind when they jumped into the iterative prototyping, rapid application **development** world of client/server computing are discovering that, like it or not, Computer-Aided Software Engineering (CASE...

...right along. Client/server versions of leading host-based CASE tools are shipping already, and more CASE **products** for client/server computing are in the pipeline. From the developers' standpoint, CASE for client/server computing...

...encourages fast, seat-of-the-pants iterative prototyping. Where host-based CASE tools tried to capture and **generate** the entire application at once in monolithic, procedural **code**, client/server computing demands small, highly modular and partitioned pieces of **code** to **produce** event-driven applications. Nevertheless, fast, iterative prototyping and seat-of-the-pants techniques can take developers only so...

...Judith Hurwitz, president of Hurwitz Consulting, Watertown, Mass., and publisher of the Client/Server Tool Watch newsletter. "More things are **happening**. There are **more** and **different** components. Developers really need tools to help them **create** a structure." Those tools are client/server CASE **products**. Effective client/server CASE **products** let developers capture the application's business requirements easily, define business **rules** and **design** the application quickly, either by drawing the components and relationships between them using graphical tools or by ...

...language and point-and-click tools. The CASE tool also must provide for bidirectional communication between the **design** and build stages, so changes made in one are reflected automatically in the other. The CASE Difference Client/server CASE tools differ from conventional CASE tools in a number of ways. First, the **design objective** of conventional CASE tools is to identify, predict and capture all the requirements of the finished application from the outset. Client/server CASE tools, by contrast, are **designed** to facilitate change, primarily through the repository and two-way information flow. Client/server CASE tools also differ from conventional CASE tools in the modularity of the **design** and **code** they **produce**. Conventional CASE tools **generate** monolithic, procedural applications that run on the host. Client/server tools **generate** modular **code** and partition the application among client, server and middleware components. Finally, client/server CASE tools differ from...

...s client/server environment is highly heterogeneous at every level. The CASE tools and the applications they **generate** must be able to run on a wide range of client and server platforms, and access a variety of relational and nonrelational databases. **Making** a CASE for C /S The earliest client/server tools were disappointing. "In a lot of cases...

...Framingham, Mass., says, "At first, CASE for client/server meant lower CASE." Lower CASE tools automate the **code generation** and construction stage of **development** but lack the business modeling and **design** component, referred to as upper CASE. A second **generation** of client/server tools, however, addresses the full range of CASE requirements. According to Hurwitz, the key components of client/server CASE **products** include: /\ Repository or knowledge base. A central source where business **rules** are articulated and the application components are managed to facilitate change and, where possible, reuse. /\ Version control and **configuration** management. A method for tracking different versions of the application and individual components, and coordinating the work of multiple developers. /\ Automatic **code generation** and partitioning. The capability to **generate** CASE **designs** and repository **objects** as working client, server and middleware **code**. /\ Two-way communication flow. Between the **design** components and build components, usually via the repository. /\ Cross-platform support, or middleware. Handles multiple clients, servers and databases across a variety of networks. /\ Published APIs. Allows **design** elements, definitions or application **objects** **created** with another tool to be incorporated into the new application. These components comprise a framework for evaluating client/server CASE tools. No **product** yet delivers all the key components well, but the following deliver enough to be considered effective client ...

...s Foundation for Cooperative Processing (FCP) is the client/server version of Andersen's conventional Foundation CASE **product**. Like the host **product**, FCP provides both upper and lower CASE capabilities and is intended for large-scale **development**. FCP **uses** Andersen's Method/1 methodology, a big, complex methodology, like traditional CASE, that walks developers through initial business information planning, **design** strategy, data modeling, project definition and more. You don't have to **use** the methodology at all, says Byron Miller, FCP developer, but he recommends that you **use** a subset of the methodology, at least. In terms of the above framework, FCP provides most of the pieces to some degree. The Foundation Repository maintains the voluminous information **generated** by Foundation's data modeling, Entity-Relationship modeling, and other modeling and diagramming **design** components. The Repository stores the information as **objects** that can be reused in other FCP applications. The information flow, however, is one-way only, from **design** to the Repository to the build component. Although the data in the Repository can be read using standard SQL, FCP's Repository is not open. Only FCP modules **make use** of the information as it is stored in the Repository, and information cannot be imported from or...

...to other tools. FCP provides a messaging architecture and Applications Programming Interface (API) to support cross-platform **development**. Foundation, however, supports only a relatively small number of clients, servers and databases (see chart on page 88). FCP also lacks **configuration** and versioning control, although Miller says it will be added in the future. Texas Instruments IEF forit delivers some of the **features** lacking in FCP, such as version control and **configuration** management for workgroups. IEF for Client/Server also partitions applications between client and server automatically. Like FCP, it is methodology-based and driven by **design** models stored in a repository called the Encyclopedia. **Designs** are then **generated** as application **code**. Subsequent changes are made to the **design**, and the system regenerates new **code**. "The auto **code generation produces** 100 percent syntactically correct **code**, but we still have to correct **design** logic errors," says Gerry Wethington, director of information systems for the Missouri State Highway Patrol, Jefferson City, Mo. All changes are made using the IEF **design** tools. The tool does not support two-way information flow between **design** and build phases. IEF supports cross-platform **development**. An alliance with PeerLogic promises to enhance IEF middleware **connectivity** capabilities. TI also promises to extend platform support significantly and provide an open API to enable other...

...Engineer differs markedly from the other two tools. Based on LBMS' conventional upper CASE tools, it provides **design** and analysis capabilities for popular 4GL client/server **development** tools, starting with PowerBuilder and soon SQLWindows and Uniface. System Engineer also differs by providing two-way communication between its repository and the **design** and build components. For example, a developer **uses** the System Engineer **design** tool to **design** an application that is then **generated** automatically for PowerBuilder. The developer can **make** changes in either PowerBuilder or System Engineer, and the changes are captured by and updated in the repository, keeping the **design** and the application in synch. "The bidirectional communication is very important to us and saves a lot of work. People here don't like to go back to the **design** tool! and **make** changes," says Philip Fasone, senior manager of **quality** assurance at Holiday Inn Worldwide Information Technologies, Atlanta. The United Nations Children's Fund (UNICEF), New York, began developing a major application with PowerBuilder before getting the bidirectional **link** to System Engineer. "Even though 25 percent of the application was already done, we were able to..."

...the information into the System Engineer repository," says Frank Buchanan, data administration and planning officer. The application **design** was then completed in System Engineer, which **generates** the basic PowerBuilder libraries automatically. The repository is the heart of System Engineer. Although built on the Gupta SQL database, the repository

is not an open platform but can be used only with the **development** tools with which it is tightly integrated, such as PowerBuilder. The repository imposes some version control on the process by managing access to **objects** and identifying groups of **objects** that together form a complete, deliverable application. System Engineer's cross-platform support is limited to the...

...more client/server CASE tool choices in the future. Although the current crop of client/server CASE **products** addresses most of the basic requirements of client/server computing, there is still plenty of room for ...

...areas of two-way communication, an open repository and cross-platform support. Of all the new CASE **products** for client/server, the most exciting twist is the way in which LBMS' System Engineer brings integrated CASE to a popular client/server application **development** tool like PowerBuilder. Leverage what you have and what you know into full-blown CASE, or ease into client/server with an application **development** tool and add CASE incrementally later on. Either approach **makes** a lot of sense. Alan Radding is a freelance writer, based in Newton, Mass., who specializes in application **development** and networking. He can be reached on CompuServe at 72320,2763.

25/5,K/3 (Item 1 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01708266 SUPPLIER NUMBER: 16209707 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Ochre brings development environment out from down under in competition with Forte Software, Dynasty and Informix. (Ochre Development Inc's Ochre Platform) (Company Profile)
Computergram International, CGI10060013
Oct 6, 1994
DOCUMENT TYPE: Company Profile ISSN: 0268-716X LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 716 LINE COUNT: 00057

FILE SEGMENT: CD File 275

IN COMPETITION WITH FORTE SOFTWARE, DYNASTY AND INFORMIX

Ochre **Development** Inc, the Australian client-server application **development** house that is **setting** -up in Burlingame, California (CI No 2,509) is pitching its Ochre Platform against what it describes as **other second generation development** system independent software vendors such as Forte Software Inc, Dynasty Technologies Inc and even Informix Software Inc...

...application on different parts of a client-server system using some fairly complex messaging. It incorporates an **object** -oriented **scripting** language, mixed graphical and command line, with all the usual **object** functions and **features** it claims can be learnt in three days. The model can already interface to Tuxedo and Top End **transaction** processing monitors with Encina support to follow. The system enables **rules** and applications **created** with its system to move freely around the system environment, on to different processors and operating systems. Existing applications can be wrapped for **use** within the Ochre environment, but remain **tied** to their system of origin. The Ochre engine is installed on the client and server and comes with a dynamic, shared repository for **object** information that the company says can be accessed at all times. Executable **code** is kept in an **object** store. The Ochre Platform server **generates** messages to be sent to other parts of the system, but those messages can ride atop a...

...passing software and says its infrastructure can be quickly tailored to provide direct access to specific Common **Object** Request Broker Architecture, **Object** Request Brokers, Distributed Computing Environment

and other Remote Procedure Calls. TCP/IP and other technologies, such as...

...be supported as requested. Indeed, one customer has integrated its Ochre Platform with ICL Plc's DAIS **Object** Request Broker. The server itself is a 600Kb executable with a Windows front-end. Ochre's dynamic model enables core business **rules** to be modified for deployment in different market or geographical areas. This presents coherency problems, the company...

...or differences that a static model cannot accommodate. For the beginning of 1995, the company plans an **object** synchroniser in Ochre Platform Release 5.0; it is currently at 4.3. Ochre is looking for...

25/5,K/4 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01668856 SUPPLIER NUMBER: 15050886 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Should you do OS/2 development? (OS/2 Developer)
Moskowitz, David
Data Based Advisor, v12, n2, p34(4)
Feb, 1994
ISSN: 0740-5200 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1866 LINE COUNT: 00147

ABSTRACT: DOS and Windows applications usually are not responsive to end users because both environments suffer from severe limitations; the DOS architecture is **designed** to keep the system busy, while building responsive applications in Windows requires the creation of an idle-time loop. Operating systems should keep the user busy; OS/2, a pre-emptive, multithreaded, multitasking operating system, is more effective at accomplishing this task than either Windows or DOS. It lets multiple programs or multiple parts of a single program execute simultaneously and works with a variety of excellent 32-bit C-language compilers. Those who do not want to learn C have several choices for OS/2 **development**. Computer **Associates** ' CA-Realizer is an OS/2 superset of BASIC; several other products **use** REXX, the native OS/2 macro language. IBM's System **Object** Model (SOM) Toolkit, while difficult to find, is the best tool for **object**-oriented **development**. Other **object**-oriented OS/2 programming products include C++ compilers and **code generators**. OS/2 supports industrial-strength databases, including IBM's DB2/2 and Computer **Associates** ' CA-dBFast.

DESCRIPTORS: OS/2; Guidelines; Application Development Software;
Programming; Database Design; Operating System; Comparison; Multitasking
SIC CODES: 7372 Prepackaged software
OPERATING PLATFORM: OS/2
FILE SEGMENT: CD File 275

...ABSTRACT: are not responsive to end users because both environments suffer from severe limitations; the DOS architecture is **designed** to keep the system busy, while building responsive applications in Windows requires the creation of an idle...

...C-language compilers. Those who do not want to learn C have several choices for OS/2 **development**. Computer **Associates** ' CA-Realizer is an OS/2 superset of BASIC; several other products **use** REXX, the native OS/2 macro language. IBM's System **Object** Model (SOM) Toolkit, while difficult to find, is the best tool for **object**-oriented **development**. Other **object**-oriented OS/2 programming products include C++ compilers and **code generators**. OS/2 supports industrial-strength databases, including IBM's DB2/2 and Computer **Associates** ' CA-dBFast.

25/5,K/5 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01663997 SUPPLIER NUMBER: 15020634 (USE FORMAT 7 OR 9 FOR FULL TEXT)
**Building a framework for object technology. (Mission Critical:
Client/Server) (Column)**
Comaford, Christine
PC Week, v11, n1, p54(1)
Jan 10, 1994
DOCUMENT TYPE: Column ISSN: 0740-1604 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 612 LINE COUNT: 00049

ABSTRACT: Object-oriented languages can be extended through subclassing using implementation inheritance, but in uncontrolled environments interface inheritance, a condition in which the objects are peers, eliminates unexpected object breaking. Applications are easier to code using implementation inheritance, but more robust systems can be created using interface inheritance. Implementation inheritance and interface inheritance are both provided by IBM's SOM. The ideal scenario is to have two independent object models, one that is easy to use, such as Microsoft Foundation Class, and one that is robust, such as Object Linking and Embedding (OLE). Identifying objects is another concern with object technology. A transaction model that allows rollback and commitment of changes should be offered by an object storage model.

DESCRIPTORS: Object-oriented programming; Software Design; Object Code;
Application Development Software
PRODUCT/INDUSTRY NAMES: 7372510 (Computer Language Software ex Military)
SIC CODES: 7372 Prepackaged software
FILE SEGMENT: CD File 275

... inheritance is an alternative in which all objects are peers.
IBM's SOM tries to provide both **implementation** inheritance via application frameworks and interface inheritance via a system **object** model. **Implementation** inheritance **makes** applications easier to **code**, while interface inheritance **makes** systems **more** robust. Unfortunately, often the **features** that give you ease of coding are the ones that adversely affect robustness. The solution is to provide two **object** models that are independent: one for ease of **use** (such as the Microsoft Foundation Class) and the other for system robustness (such as **Object Linking and Embedding**).

Identifying objects is another challenge. With Apple's OpenDoc, objects have ASCII names. This makes objects more...

25/5,K/6 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01610659 SUPPLIER NUMBER: 14084011 (USE FORMAT 7 OR 9 FOR FULL TEXT)
**Object Software launches OST-Look. (Object Software Technology Ltd.'s
animation software) (Product Announcement)**
Computergram International, CGI07120014
July 12, 1993
DOCUMENT TYPE: Product Announcement ISSN: 0268-716X LANGUAGE:
ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 306 LINE COUNT: 00026

FILE SEGMENT: CD File 275

TEXT:

Objective Software Technology Ltd, Livingston, Scotland, will introduce OST-Look, an animation tool **designed** to enable users to reconstruct an animated **object** -level view of a C++ program as it's executed. The firm, **set up** by two ex-British Telecommunications Plc C++ software engineers using funds from the government-run Scottish Enterprise ...

...is a graphical C++ monitoring system that runs under Microsoft Windows, NT and OSF/Motif. According to **Objective** Software Technology, the software extracts **objects** and class **design** -levels embodied in C++ programs and **uses** them to **generate** active **object**, class and message diagrams. Traditionally, most developers have just used debuggers, **setting** breakpoints when specified lines or functions are executed, to **generate** **code** information, says Eddie Anderson, **Objective**'s managing director. But OST-Look adopts a different approach. It **uses** the structure of **object**-oriented programs to select specified parts of **code** and represents the application graphically on screen. Multiple windows can be **created** to run up to five different views of a program simultaneously. Other OST-Look **features** include: graphical replay, which lets users replay **sequences** of an application after it has been run; source viewing, which provides source-level debugging and editing capabilities; and data viewing, which enables multiple data browsing and editing windows to be **created** from an **object** layout. The software is available now for Sun and Windows systems - Windows NT versions will follow later in the year. **Objective** is now developing an **object**-oriented testing tool for third party software engineering development environments, due third quarter next year.

25/5,K/7 (Item 5 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01605227 SUPPLIER NUMBER: 14006721 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Lotus pushes development platform for Notes into beta. (Lotus Notes)
Rooney, Paula
PC Week, v10, n26, p1(2)
July 5, 1993
ISSN: 0740-1604 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 439 LINE COUNT: 00035

ABSTRACT: Lotus **Development** Corp is beta testing a new **object**-based **development** environment for its Notes office-automation platform that will let users **create** applications that access and integrate information from a variety of databases. The **product** is **code**-named Notebook and will reportedly **use** pre-**configured objects** to build applications. Users can view and **link** data from SQL Server and other databases within Notes and integrate the information with that in Notes' own database. Notebook **uses** the DataLens application programming interface to translate proprietary formats and provide **connectivity** to more than 40 databases; it offers a Windows front-end program with tool bars and **other** convenient interface **features**. Notebook will reportedly be released in the fall of 1993.

SPECIAL FEATURES: illustration; table
COMPANY NAMES: Lotus Development Corp.--Product development
DESCRIPTORS: Application Development Software; Product Development
SIC CODES: 7372 Prepackaged software; 7373 Computer integrated systems design
TICKER SYMBOLS: LOTS
TRADE NAMES: Lotus Notes (Workgroup software)--Computer programs
OPERATING PLATFORM: MS Windows
FILE SEGMENT: CD File 275

ABSTRACT: Lotus **Development** Corp is beta testing a new **object**-based **development** environment for its Notes office-automation platform that will let users **create** applications that access and integrate information from a variety of databases. The **product** is **code**-named Notebook and will reportedly **use** pre-**configured objects** to build applications. Users can view and **link** data from SQL Server and other databases within Notes and integrate the information with that in Notes' own database. Notebook **uses** the DataLens application programming interface to translate proprietary formats and provide **connectivity** to more than 40 databases; it offers a Windows front-end program with tool bars and **other** convenient interface **features**. Notebook will reportedly be released in the fall of

1993.

25/5,K/8 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01545586 SUPPLIER NUMBER: 12858531 (USE FORMAT 7 OR 9 FOR FULL TEXT)
**Clarion Software Corp. (Clarion Database Developer 3.0 database program
development tool) (Comdex News) (Brief Article) (Product Announcement)**
Data Based Advisor, v10, n11, p116(1)
Nov, 1992
DOCUMENT TYPE: Product Announcement ISSN: 0740-5200 LANGUAGE:
ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 111 LINE COUNT: 00009

COMPANY NAMES: Clarion Software Corp.--Product introduction
DESCRIPTORS: Application Development Software; Product Introduction
SIC CODES: 7372 Prepackaged software
TRADE NAMES: Clarion Database Developer 3.0 (Database application
development software)--Product introduction
OPERATING PLATFORM: MS-DOS
FILE SEGMENT: CD File 275

TEXT:

...Software will demonstrate Clarion Database Developer Version 3, a major upgrade of its DOS-based database application **development tool**. Version 3 **uses** the TopSpeed optimizing **code generator**. Other key **features** include replaceable database drivers (RDDs) to permit applications to access a variety of databases, visual template and **embedded source code** enhancements to the application **generator**, automatic overlays and dynamic **link** libraries (**DLLs**) to run large application programs within the DOS limit of 640K, mixed graphics and text, mouse support...

25/5,K/9 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01534848 SUPPLIER NUMBER: 12611731 (USE FORMAT 7 OR 9 FOR FULL TEXT)
**What is LightShip? (Pilot Software Inc.'s LightShip 3.01 executive
information system tool kit and graphical user interface builder)**
Job, Mark
Data Based Advisor, v10, n8, p62(1)
August, 1992
ISSN: 0740-5200 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1321 LINE COUNT: 00106

ABSTRACT: Pilot Software Inc's LightShip 3.01 is a graphical user interface (GUI) builder and executive information system (EIS) toolkit that accelerates application development and uses a limited **object** -oriented design. Developers build applications in LightShip by drawing **objects** on a screen, initializing attributes of **objects** and attaching data queries and application-specific code. LightShip can also infer logic from **object** relationships and variables, making it unique when compared with **other** visual **development** programs such as ToolBook or VisualBasic. A Windows dynamic **link** library (**DLL**) called Lens is an add-on **product** developers can invoke by clicking on a document **object**. The Lens Browse mode lets users access data dictionary information to retrieve and display data without having to know a database's **syntax rules**. Advanced users can request data using more complex Structured Query Language (SQL) queries.

COMPANY NAMES: Pilot Software Inc.--Products
DESCRIPTORS: Executive Information Systems; Applications Programming; GUI
SIC CODES: 7372 Prepackaged software

TRADE NAMES: LightShip 3.01 (Executive information system)--Design and construction

OPERATING PLATFORM: MS Windows

FILE SEGMENT: CD File 275

...ABSTRACT: interface (GUI) builder and executive information system (EIS) toolkit that accelerates application development and uses a limited **object**-oriented design. Developers build applications in LightShip by drawing **objects** on a screen, initializing attributes of **objects** and attaching data queries and application-specific code. LightShip can also infer logic from **object** relationships and variables, making it unique when compared with **other** visual **development** programs such as ToolBook or VisualBasic. A Windows dynamic **link** library (**DLL**) called Lens is an add-on **product** developers can invoke by clicking on a document **object** . The Lens Browse mode lets users access data dictionary information to retrieve and display data without having to know a database's **syntax rules** . Advanced users can request data using more complex Structured Query Language (SQL) queries.

25/5,K/10 (Item 8 from file: 275)

DIALOG(R) File 275:Gale Group Computer DB(TM)

(c) 2003 The Gale Group. All rts. reserv.

01517886 SUPPLIER NUMBER: 12149718 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Software as building blocks. (Column)

Miller, Michael J.

PC Magazine, v11, n11, p81(2)

June 16, 1992

DOCUMENT TYPE: Column ISSN: 0888-8507 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 1092 LINE COUNT: 00086

ABSTRACT: Software developers are increasingly using reusable code when creating new applications. **Object**-oriented programming is gaining popularity as more languages such as C++ are being used. **Object**-oriented programming enables users to reuse blocks of code to build new applications. This may become the trend in software **development** as **object**-oriented programming and **other** techniques, such as **object** linking and **embedding** (OLE), become more popular. Ultimately, users should be able to **create** their own applications or customize existing ones so that their software better suits their work habits. The current trend of software **development** dictates that a full- **featured** package has a marketing advantage over a single-purpose or simpler package. Customization and reusable- **code implementation** techniques will ensure that software packages retain some user friendliness.

DESCRIPTORS: Critique; Industry Analysis; Reusable Code; Outlook;

Object-oriented programming; Customization; Program Development Techniques

SIC CODES: 7372 Prepackaged software

FILE SEGMENT: CD File 275

ABSTRACT: Software developers are increasingly using reusable code when creating new applications. **Object**-oriented programming is gaining popularity as more languages such as C++ are being used. **Object**-oriented programming enables users to reuse blocks of code to build new applications. This may become the trend in software **development** as **object**-oriented programming and **other** techniques, such as **object** linking and **embedding** (OLE), become more popular. Ultimately, users should be able to **create** their own applications or customize existing ones so that their software better suits their work habits. The current trend of software **development** dictates that a full- **featured** package has a marketing advantage over a single-purpose or simpler package. Customization and reusable- **code implementation** techniques will ensure that software packages retain some user friendliness.

25/5,K/11 (Item 9 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01501148 SUPPLIER NUMBER: 11971981 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Microsoft aims for object leadership with C/C++ 7.0 development system for Windows. (Product Announcement)
Computergram International, n1868, pCGI02280006
Feb 28, 1992
DOCUMENT TYPE: Product Announcement ISSN: 0268-716X LANGUAGE:
ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 863 LINE COUNT: 00071

COMPANY NAMES: Microsoft Corp.--Product introduction
DESCRIPTORS: Product Introduction; Application Development Software;
Object-oriented programming
SIC CODES: 7372 Prepackaged software; 3577 Computer peripheral
equipment, not elsewhere classified
TRADE NAMES: Microsoft Windows 3.0 (GUI)--Computer programs; Microsoft
C/C++ 7.0 (Compiler)--Product introduction
OPERATING PLATFORM: MS Windows
FILE SEGMENT: CD File 275

... 0 DEVELOPMENT SYSTEM FOR WINDOWS

Microsoft Corp claims that its new Microsoft C/C++ version 7.0
Development System for Windows, with its High-performance **Object**
Technology, enables **development** of the smallest, fastest Windows-based
applications possible. The **product** contains a comprehensive set of
professional tools for Windows 3.1, so developers can start work on
applications for the new release immediately. The High-performance **Object**
Technology is a set of integrated technologies for C++ **design**, new
compiler optimisations, and a Windows-based Applications Framework to
maximize performance of **objects** under Windows, and is claimed to provides
the benefits of **object** -oriented **design** without the high memory overhead
and slow performance of **other C++ development products**. C/C++ 7.0 has
a complete Windows 3.0 and Windows 3.1 tool set, including the Windows
debug kernel and the Windows **set - up** toolkit, a new version of CodeView
for C++ debugging, a faster Programmer's Workbench, and the Qualitas 386MAX
memory manager to ease memory constraints during **development**. C/C++ 7.0
is intended to handle large, complex Windows-based applications, and
Microsoft claims that robust C++ version 2.1 **code generation** avoids
many of the errors experienced by other C++ compilers while conforming
strictly to the AT&T Co C++ 2.1 specification, and is **designed** "to
generate correct **object code** from even the most complex or obscure
source expressions". Foundation Classes provide **objects** for Windows, with
more than 60 C++ classes that abstract the functionality of the Windows
Application Programming Interface - and the full Interface is supported.
There are classes for the GDI Windows graphics system, **Object Linking**
and **Embedding** and menus. The framework is intended to provide easy
migration from the procedural programming methodology of C and the Windows
Programming Interface to the **object** -oriented approach of C++. Developers
can add **object** -oriented **code** while retaining the ability to call any
Windows Programming Interface function directly at any time; a programmer
...

...application from scratch. The foundation classes also simplify Windows
message processing and other details programmers must otherwise **implement**
manually, and they include extensive diagnostics. They have undergone
"rigorous" tuning and optimisation to yield very fast...

...Foundation class library occupies only 39Kb and should add very little
overhead to the application. C++ source **code** is included for all
foundation classes.

20,000 lines of sample code
There are also over 20...

25/5,K/12 (Item 10 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01463576 SUPPLIER NUMBER: 11613732 (USE FORMAT 7 OR 9 FOR FULL TEXT)
News from Comdex. (Comdex/Fall '91)
Gutierrez, Dan; Gillespie, Kelly; Litwin, Paul; Nesbitt, Kenn;
Schaffhauser, Dian
Data Based Advisor, v9, n12, p146(3)
Dec, 1991
ISSN: 0740-5200 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2898 LINE COUNT: 00228

ABSTRACT: Highlights from the Comdex/Fall 1991 trade show include the strong presence of Microsoft Windows. Microsoft Corp provides exhibits devoted to Windows Multimedia, Windows for Pens and other Windows applications as well as demonstrations of its Windows NT (New Technology). Windows for Pens is expected to be released early in 1992 and will let standard Windows programs accept handwritten and symbolic input from a stylus attached to the computer. Apple shows its SoftPC emulator which allows Macintoshes to run DOS programs. Apple is also promoting its new hardware including the PowerBook 170 and its high-end Quadra business machine based on the 68040 Motorola chip. Novell Inc shows off its Netware Lite low-cost network operating system that is targeted at first-time microcomputer network users. Details on other products demonstrated at Comdex are presented.

COMPANY NAMES: Apple Computer Inc.--Products; Microsoft Corp.--Products; Novell Inc.--Products
DESCRIPTORS: Comdex-Fall; Trade Show; Pen-Based I/O; Emulators; Network Operating System
SIC CODES: 3571 Electronic computers; 5045 Computers, peripherals & software; 7372 Prepackaged software; 7373 Computer integrated systems design
TICKER SYMBOLS: AAPL; MSFT; NOVL
TRADE NAMES: Microsoft Windows (GUI)--Computer programs; SoftPC (Emulator)--Design and construction; NetWare Lite (Network operating system)--Design and construction
FILE SEGMENT: CD File 275

... sales of Microsoft's SQL Server, a formidable opponent to Interbase?

Borland also gave backroom demos of **ObjectVision** 2.0. From the looks of it, this appears to be the quickest maturation in a **product** ever. In about seven months, the **product** has gone from "a nice idea" to a very useful front-end tool. It's still not dWindows (talk about hyped-up ads!), but this \$149.95 Windows **product** may be one of the neatest GUI front-ends around. While not suited for large-scale data manipulation, **ObjectVision** appears well-suited for **creating** applications based on business **rules** ("decision trees" in Borland parlance) that require access to Paradox, dBASE, and Btrieve data. It also **makes** for a dandy forms **designer** tool. Besides offering numerous sample applications (including games like Jeopardy and Hang-man), the new release supports free runtime one-to-many data structures and multi-record display; **event** trees (allowing users to define actions such as database updating, modifying menus, moving to **another** form, and the like); **property** inspectors (a new Borland component enabling users to right-click the mouse to receive details about an **object** on the screen); and support for Dynamic **Link** Libraries (allowing users to **create** specialized functions or **use** those written by a third-party) and OLE support (this Windows 3.1 announced **feature** lets users access and modify **objects** from other applications without physically leaving the first one). Call (408) 438-5300.

Novell, Inc. took the...

DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01355191 SUPPLIER NUMBER: 08457574 (USE FORMAT 7 OR 9 FOR FULL TEXT)
**Information Builders builds strategy for the 1990's on gaps left by IBM's
Repository....**
Ring, Katy
Computergram International, n1428, CGI05180011
May 18, 1990
ISSN: 0268-716X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 581 LINE COUNT: 00048

COMPANY NAMES: Information Builders Inc.--Product development
DESCRIPTORS: SAA; Product Development; System Design; Multivendor Systems
; Interoperability
SIC CODES: 7372 Prepackaged software
TRADE NAMES: Focus (Database application development software)--Product
enhancement
FILE SEGMENT: CD File 275

... for throughout the decade. This means that its Focus will have to
be more than a fourth **generation** language and will evolve into a complete
tool for information management. Last month the company made two...
...integrated its Level 5 expert system with the Focus language to enable
developers to write applications containing **embedded** knowledge. Secondly,
Information Builders' new 5.5 release of PC/Focus started shipping: the
thing offers a database management system architecture called Direct/
Connect between personal computers and mainframes. This latest release of
PC/Focus fulfils Phase 1 of Information Builders' long-term Focnet strategy
to **implement** client-server co-operative processing unveiled last year in
Cannes (CI No 1,300). Next year Focus procedures will be distributed and
stored as executable **objects** and will be accessed via remote procedure
calls for cross-machine **object** execution. Phase 3 of the strategy is on
schedule for delivery in 1992 when Focus will be...

...distributed applications. He doesn't think that IBM's distributed
database will arrive this century. As for **other developments** about to
be launched on an unsuspecting world, Information Builders has just
released its Level 5 **Object product**, taking Focus into the graphical
user interface environment and offering portability across graphics
environments. However, the company...

...are too complicated for databases to handle properly. Similarly, he
doesn't have too much time for **object** -oriented techniques, saying that
while the press and theoreticians love them, he's not so sure about...
...He suggests half-teasingly that users can get a "dosage" of such
techniques by using Level 5 **Object** which will "tell you more than you
ever want to know about using that particular paradigm". Focus...

...a little further ahead to the autumn, Information Builders will have
Presentation Manager Focus available: this will **use object** -oriented
techniques by **applying** the Focus language to discrete screen **objects**,
thereby **producing** modular, reusable **code**. At the same time the company
will introduce the Focus Application Creation Tool in a version which can
be used to **generate** Focus **code** by picture programming. - Katy Ring

25/5,K/14 (Item 12 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01314366 SUPPLIER NUMBER: 07925310 (USE FORMAT 7 OR 9 FOR FULL TEXT)
**Concurrent Computer adapts Micro Focus Cobol, tools for its OS/32,
real-time Unix. (Concurrent Computer's CoDE program) (product
announcement)**
Computergram International, n1316, CGI11300009
Nov 30, 1989

DOCUMENT TYPE: product announcement ISSN: 0268-716X LANGUAGE:
ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 322 LINE COUNT: 00026

COMPANY NAMES: Concurrent Computer Corp.--Product introduction
DESCRIPTORS: Product Introduction; COBOL; Application Development
Software; Programming Language; Performance Improvement
SIC CODES: 7372 Prepackaged software
TICKER SYMBOLS: CCUR
TRADE NAMES: CoDe (Program development software)--Product introduction
FILE SEGMENT: CD File 275

FOR ITS OS/32, REAL-TIME UNIX

Concurrent Computer Corp has launched a Cobol **development** software package called **CoDE** which it claims can improve the efficiency and performance of Cobol applications by **more** than 75%, while cutting **development** time. **CoDE** is based on the Micro Focus Cobol **product** set including as it does a Cobol compiler and interpreter, Animator symbolic debugger, Forms-2 package, a...

...tools, but it differs from it in three main ways: firstly Concurrent had to write a new **code generator** for its chip set, translating the Micro Focus Cobol intermediate **code** into Concurrent's **object code**. Secondly changes had to be made via the pre-processor to the compiler itself to provide an...

...time changes have been made so that the Micro Focus symbolic debugger runs within Concurrent's Reliance **transaction** processing package. **CoDE**'s performance is achieved because it compiles and debugs in intermediate **code**, rather than having to go into binary to reach the hardware. The intermediate **code** can be interpreted directly providing a quick turnaround, since **code** can be compiled and seen executing. In this way run-time errors or logic errors can be spotted in the intermediate **code** by running Micro Focus' Animator which displays source **code** line by line on screen. In other words, **CoDE** has a fully functional symbolic debugger enabling the programmer to execute one part of the **code**, highlight it and see the output beside it. **CoDE** is **designed** for real-time applications such as securities and trading systems, banking, health-care, government services and insurance...

...system as well as under real-time Unix and is available direct from Concurrent. There is a **tiered** pricing structure for the set depending on the processor the developer **uses**.

25/5,K/15 (Item 13 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01305542 SUPPLIER NUMBER: 07742771 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Utilities and languages. (listings of software programs) (directory)
DG Review, v7, n1, p4(7)
Summer, 1989
DOCUMENT TYPE: directory ISSN: 1050-9127 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 8309 LINE COUNT: 00734

ABSTRACT: A directory of utility and language software packages, grouped by specific applications, is given. The program listings include product name and price, vendor name, a brief description of the product and information concerning the product's hardware, operating system and language requirements. The categories listed include communications, fourth-generation languages, operating system enhancements, program development aids, program generators, programming languages, report generators, screen editors-formatters and utilities.

DESCRIPTORS: Directories; Desktop Utility; Fourth-Generation Language;

Operating System; Application Development Software; Program Generators;
Programming Language; Report Generation Software; Vendors; Screen
Generators/Formatters
SIC CODES: 7372 Prepackaged software
FILE SEGMENT: CD File 275

... dictionary, which integrates the language's three modules: QUICK,
an interactive screen generator, online data entry and **transaction**
processor; QUIZ, an online query tool and report writer; and QTP, a
powerful volume processor. PowerHouse's built-in defaults and high-level
statements spur prototype **development**. Custom constructs and **additional**
functions can be added to transform prototypes into **production**
applications. PowerHouse meets the user's level of expertise; it provides
the advanced capabilities demanded by experienced users and allows users
without technical training to take an active role in application
development. Price: Contact vendor CPU: Eclipse MV/Family Operating
Systems: AOS/VS Languages: ICOBOL, INFOS, DG/SQL Reference Number: 291
Contact: Dan Scharf Telephone: (613) 738-1440 INFODEX Application:
Recreation of ICOBOL source **code** **Product** Name: ReSource Description:
ReSource recreates ICOBOL source **code** from existing **object** files (.DO
and .PO). Price: Contact vendor CPU: Desktop, MicroEclipse, Eclipse,
MV/Family Operating Systems: RDOS, AOS...

...Klein Telephone: (201) 662-7020 MAXON COMPUTER SYSTEMS, INC.
Application: Compatible high-performance replacement for Business BASIC
Product Name: B32 Description: B32 is a completely compatible 32-bit
implementation of Data General Business BASIC. It **features** almost
doubled performance, program size up to 512KB, up to 32 channels and many
other extensions and new **features**. By taking advantage of the full
32-bit **instruction** set, system overhead is significantly reduced. All
programs written under Business BASIC will operate identically. Conversion
is...

...BASIC Reference Number: 696 Contact: Brenda Allen Telephone: (416)
445-4823 TRANSOFT, LTD. Application: Portable Business BASIC **Product**
Name: UBB Universal Business BASIC Description: UBB is the first Business
BASIC look-alike which is portable to UNIX, MS-DOS and DOS networks. BBASIC
source compatibility is provided with special **features** for directory and
link management, screen and keyboard **attributes**, and special system
calls. UBB also retains 100% compatibility with BBASIC data files, so no
data file conversion or **set - up** is necessary when porting live systems.
Extra UBB **features** include 1MB+ programs, 65KB line numbers, 65KB
variables, 256 channels, hot keys, windows and intelligent help. Also
available with UBB is Transoft's new structured query language--U/SQL (see
Report **Generators**). Price: Contact vendor Operating Systems: UNIX,
MS-DOS, DOS networks Languages: Business BASIC Reference Number: 946
Contact: Mike Edwards Telephone: +44 753 692332 WILD HARE COMPUTER SYSTEMS,
INC. Application: Programming languages **Product** Name: Choice! Description:
Choice! directly runs Data General Interactive COBOL (ICOBOL) programs on
non-data General computers. The actual program **object** files and data
files from the Data General machines are used on all other computers,
thereby ensuring...

...Data General data file formats, including ISAM files with alternate
keys. All of the special ICOBOL "extended" **features**, such as the SCREEN
SECTION, CALL PROGRAM system calls, program **LINKAGE** section and menu
input handling, are exactly duplicated. Price: Contact vendor CPU: IBM,
NCR, AT&T, DEC...

...Number: 883 Contact: Sales Department Telephone: (303) 442-0324 WILD
HARE COMPUTER SYSTEMS, INC. Application: Language compiler **Product** Name:
Prism Description: Wild Hare's COBOL compiler allows ICOBOL programs to be
compiled on different types...

...computers, from small PCs to large mainframes. This fast compiler
translates ICOBOL source programs into DG-compatible **object code** at
speeds greater than 7,000 lpm on typical PCs. Since the **produced object**

code is compatible with DG's ICOBOL run-time systems, the resulting program can be run on any...

25/5,K/16 (Item 14 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01284244 SUPPLIER NUMBER: 07209533 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Software development: tech releases. (product announcement)
PC Tech Journal, v7, n1, p41(2)
Jan, 1989
DOCUMENT TYPE: product announcement ISSN: 0738-0194 LANGUAGE:
ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 820 LINE COUNT: 00068

DESCRIPTORS: Program Editor; Computer programs; Programming Language;
Compiler; Windowing; Computer-Aided Software Engineering
FILE SEGMENT: CD File 275

TEXT:

SOFTWARE DEVELOPMENT For machines running DOS, Xenix, OS/2, FlexOS, and CP/M-86 operating systems, CompuView **Products** has announced an enhanced version of its VEDIT PLUS editor. Version 3.0 includes pull-down menus, a multilevel Undo (1,000 levels), a full **implementation** of Unix-style regular expressions, columnar blocks, context-sensitive help, and a pop-up ASCII table. As...

...from within the editor. Fully definable keystroke assignments allow the user to customize the editor. \$185. CompuView **Products** Inc., 1955 Pauline Blvd., Suite 300, Ann Arbor, MI 48103; 313/996-1299 A computer-aided systems engineering (CASE) **product** that **produces** C source-**code** and executable programs directly from graphics specifications has been introduced by SYSCORP International. Using MicroSTEP's mouse-driven, graphics specification environment, a developer interactively **creates** a system specification with sets of **design** tools **designed** to build data-flow diagrams, specify the data structures, lay out the screens, format the reports, and describe the application's computations and processing logic. Elements of a **design** specification can be copied and stored in the data dictionary for **use** in other specifications. \$5,000. SYSCORP International, 9420 Research Blvd., Suite 200, Austin, TX 78759; 800/727...

...512/338-0591 An OS/2 version of Polytron's programming aid PolyAWK is shipping. A powerful **pattern**-matching language for writing short programs to perform common text-manipulation tasks, PolyAWK contains many **features** of the C programming language. **Other features** of PolyAWK include text substitution, definable functions, and built-in functions for strings, integers, and floating-point...

...s QuickBASIC has begun shipping. Version 4.5 includes QB Advisor, an on-line reference system that **uses** hypertext technology. Two other additions, QB Menus and QB Express, help first-time users become comfortable and **productive** in the QuickBASIC environment. Offering the same **features** as QuickBASIC 4.0, version 4.5 includes a smart **syntax**-checking editor, rapid compilation at speeds as fast as 150,000 lines per minute, and an improved...

...superset of ANSI C that supports the latest ANSI C extensions while preserving compatibility with existing C **code** is available from Stepstone. The **Objective**-C Language 4.0 compiler **features** extensive syntactical and type checking to detect common programming errors. Type checking includes full ANSI C function...

...prototyping for classes, and type-mismatch detection for all data-function combinations as well as for all **objects** with types that are declared by the programmer, even if the programmer is not using an

ANSI-compatible C compiler. **Other features** include enhanced error handling, a message-tracing facility, and a wide range of binding options. \$495. The...

...Systems that offers programmers an automatic way of managing any size software project has been released. MKS **Make** automatically updates files and speeds up other tasks, such as documentation **production**, directory clean-up, software installation, and administration. AR, an **object - code** librarian program that has the look and feel of Unix AR, but is compatible with Microsoft LIB, is included with the package. MKS **Make** provides full compatibility with Unix System V AUGMake under DOS and user-definable **meta-rules** and dependencies, as in Sun OS **Make**. Support for **linkers**, libraries, revision-control systems, and a wide range of compilers is provided. \$149. Mortice Kern Systems Inc...

25/5,K/17 (Item 15 from file: 275)
DIALOG(R) File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01031555 SUPPLIER NUMBER: 00647623

How to Choose a C Compiler.

Hunt, B.

PC Tech Journal, v1, n1, p78-80

July-Aug., 1983

DOCUMENT TYPE: buyers guide

ISSN: 0738-0194

LANGUAGE: ENGLISH

RECORD TYPE: ABSTRACT

ABSTRACT: The **use** of C programming language on an IBM PC is an alternative to BASIC because it is practical and efficient. A **quality** C compiler will **produce** a level of performance close to those of large computers. A good compiler requires 128K bytes of RAM and two double sided drives with 360K bytes capacity. Factors to examine when choosing a C compiler are size of compiler disk files, compiler speed, error messages, compatibility with DOS, support of C language, included library functions, interface with **assembler** functions memory usage and performance comparisons with bench marks. **Other** criteria to examine are overall **quality** (documentation), support of separate compilation, and **production** of **object code** ready for **linking**. When purchasing a compiler certain costs must be expected and time must be taken to evaluate different compilers available. Included are tables listing selection criteria for C compilers, disk drive **setup** for a C compiler and library functions. An index of available C compilers is included.

SPECIAL FEATURES: illustration; table

DESCRIPTORS: C Programming Language; Criteria; Compiler; Personal Computers; Disk Operating System; Tutorial

OPERATING PLATFORM: Disk Operating System

PROGRAMMING LANGUAGE: C Programming Language

FILE SEGMENT: CD File 275

ABSTRACT: The **use** of C programming language on an IBM PC is an alternative to BASIC because it is practical and efficient. A **quality** C compiler will **produce** a level of performance close to those of large computers. A good compiler requires 128K bytes of...

...files, compiler speed, error messages, compatibility with DOS, support of C language, included library functions, interface with **assembler** functions memory usage and performance comparisons with bench marks. **Other** criteria to examine are overall **quality** (documentation), support of separate compilation, and **production** of **object code** ready for **linking**. When purchasing a compiler certain costs must be expected and time must be taken to evaluate different compilers available. Included are tables listing selection criteria for C compilers, disk drive **setup** for a C compiler and library functions. An index of available C compilers is included.

25/5,K/18 (Item 1 from file: 636)
DIALOG(R) File 636:Gale Group Newsletter DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01773831 Supplier Number: 42953742 (THIS IS THE FULLTEXT)

Principles of Object-Oriented Design

Financial Technology Insight, pN/A

May, 1992

ISSN: 0961-5342

Language: English Record Type: Fulltext

Document Type: Newsletter; Trade

Word Count: 1933

TEXT:

Object -oriented programs consist of a **collection** of interacting **objects** that are instances of classes. **Object** -oriented **design** is based upon two essential concepts: encapsulation and inheritance. Encapsulation refers to the principle of storing both data structures and the process (known as 'methods' in OOD parlance) that act upon them as indivisible entities, or **objects**. In its behaviour, therefore, an **object** is just like a very simple computer. It is capable of acting on **instructions** ('messages'), and of sending them to other **objects**. Messages are the only means of access to the data **associated** with an **object**. They specify what operation is required on the data, but not how it is **implemented**. Both the data and methods of a given **object** are private, and it is this which gives **objects** and their operations their inherent modularity. In many respects, **objects** are simply another instance of the information hiding principles first embodied in the concept of abstract data types. A number of conventional programming languages already support the specification of abstract data types, but they lack the concept of inheritance. Each **object** is a member of a specific class, and this defines its **properties**. Classes are related to one **another** through inheritance. Aggregation and generalization are forms of inheritance which permit existing **object** classes to be grouped together to **produce** a new **object** class. For example, the **object** class customer might be defined as an aggregation of the classes current account, deposit account, mortgage etc., whilst the class **employee** might be defined as the generalization of the classes branch manager, teller, ledger clerk etc. Specialization is the reverse process of generalization, **creating** a new class by adding **properties** to an existing one, known as the parent class. Inheritance may be strict, in which case instances of a class must satisfy all the **properties** of its parent. By contrast, default inheritance allows instances of a class to contradict some **properties** of its parent. A further extension is multiple inheritance, in which a new **object** class may be a specialization of more than one parent. OOD typically begins with an informal description of the solution strategy. From this, the next step is to isolate **objects** and operations. **Objects** are identified by the noun **occurrences** within the description, and represent the information domain. Operations, which are identified by verb **occurrences**, represent the processing **associated** with the information domain. The identification of **objects** and their **attributes** is the foundation for the creation of the system's data abstractions. Consideration of **objects** and operations together leads to the specification of the system's structure. The relationships between **objects** and operations leads to the definition of how the **objects** are used, and their interfaces, i.e. the messages that are passed between them. - Impact on Software Development Productivity

Object -oriented software offers a number of potentially powerful **features** for the **design** of complex systems, and for the management of changes in data, and system functionality, over long periods of time. Proponents of OOD argue that one of the problems with conventional software **design** is that data and processing are regarded as two quite separate perspectives of the **design** task; OOD offers a way to re-integrate them. Encapsulation is a tool with great potential. In theory, it means that it is not necessary to consider other **objects** in the system when considering the new specific **object**. To **use**, test or re-**use** an **object**, it should only be necessary to understand its interface, i.e. the messages that can be sent to that **object**, and what they do. Similarly, when extending the

implementation of a particular object, it should be possible to ignore all other objects. The question of why software production has failed to follow the pattern set by hardware is a vexed one. It has been estimated that only 15% of new code actually serves a new purpose; the rest merely duplicates what many others have written before. Software libraries have been around along time, but apart from one or two exceptions (e.g. mathematical routines), have made little impact on commercial software development practice. In OOD, the problem of re-usability is identified with the lack of abstraction in conventional software design. OOD aims to raise the level of abstraction of software design through the exploitation of object classes and inheritance. A well designed class hierarchy enables the software engineer to create new objects by building upon the properties inherited from an existing class. It is then only necessary to define how a new object differs from that class. With OOD techniques, many believe that the goal of building systems from 'software ICs', selected 'off the shelf' from catalogues of objects, will finally be achieved. In summary, claims that OOD improves software productivity rest on four basic premises: 1. Objects and classes reduce the difficulties of translating complex real world situations into systems. In conventional programming languages, the designer must map the problem domain onto predefined data and control structures. In OOD, the designer can create objects and classes which closely match those in the problem domain. It is argued that inheritance is a powerful tool for capturing data semantics, and that OOD reduces the conceptual gap between the real world and its computer model. 2. Object-oriented code is flexible, and so permits rapid responses to changing user requirements. Inheritance implicitly encourages code re-use, and so helps economize on implementation and maintenance effort. 3. The re-use of standard components reduces development time and reduces the code size. A company can build up libraries of classes tailored to its specific needs and use them over and over again. 4. Maintenance costs are reduced. Information hiding means that design is decoupled from implementation, and changes are effectively isolated. In conventional software, where data and procedures are packaged separately, changes in data structure may affect many different modules, which in turn may have been written by many different programmers. In OOD, since data and procedures are packages together, the implications of most changes can be easily identified, because they are so localized. The Software Engineering Institute at Carnegie Mellon University has defined five software development 'maturity levels', which range from 'chaotic happy hacking' at level one, to the application of a systemic process with continuous self improvement at level five. According to surveys, most software development organizations are still below level two on this scale. OOD promises to help improve the maturity level of software development. The naturalness of the object notion facilitates moving smoothly from requirements via analysis and design to an implementation. In addition, it is claimed that object-oriented analysis yields qualitatively better models than structured analysis modelling. This should result in a deeper understanding of the requirements of systems and hence will reveal fatal misunderstandings earlier. Hitherto, the impact of OOD has been restricted by the limited availability across a wide range of operating environments, the importance and difficulties of integrating with existing systems, the lack of support for large scale development both in terms of methodologies and tools and the shortage of appropriate skills. Now, however, some of these obstacles are beginning to diminish. - Practical Experience with Object-Oriented Design Software engineers are entitled to treat claims that OOD will transform software development with some scepticism. There is now enough evidence from experience in the field, however, to begin to evaluate their legitimacy. Studies of the impact of object-oriented techniques on software maintenance have provided some encouraging results. Compared with a system implemented in a conventional programming language, object-oriented code was found to require fewer module changes, fewer source line changes, and fewer source lines added. Another study concluded that OOD may cut maintenance effort by as much as 50%. There is also confirmation for code re-usability promoting gains in productivity. In some cases, reductions of 5:1 in the amount of code required for a new application have been reported, together with a

substantial reduction in the size and complexity of the resultant **code**. OOD techniques have been found to raise their own **design** problems, however. A simple example is whether some part of an **object** should be represented as an instance variable, or inherited as a component class. Decisions of this kind may be of trivial importance in small scale systems and at the early stages of larger ones. Unfortunately, the correct choice may only be arrived at through trial and error, by which time it may be difficult to **implement** changes. The view persists that the key to maintenance is good **design** rather than the specifics of the tools. A large **object**-oriented system without a proper **design** foundation may be just as unmaintainable as a traditional pile of Fortran or Cobol 'spaghetti'. In commercial environments, the temptation to **produce code** quickly is the same no matter what tools are available. Conceivably, it is easy to **produce** hard to read and hard to maintain **object**-oriented **code**. The longevity of **object**-oriented **code** then reveals its disadvantages - badly written **code** propagates. It has been argued that **features** such as inheritance may actually compromise the requirements of large scale software projects. For example, the integration of software developed under different type hierarchies is very difficult. Inheritance may also lead to disadvantages when modifying programs. For example, when modifying a class, it is necessary to check all subclasses. In this way, OOD may lead to the inadvertent introduction of dependencies, which of course is bad for maintainability. There are clearly many unresolved issues surrounding OOD. From a purely practical standpoint, there is also the question of how costly and difficult the changeover to OOD would be. Opinions differ over whether the best results can be obtained by judiciously mixing OOD with **more** conventional software **development** methodologies. Clearly, anything which might reduce the potential upheaval would be highly desirable. -

Integration with Conventional Methodologies The possibility of integrating OOD with conventional methodologies is under investigation, but it is questionable whether the full benefits of OOD can be achieved in this way; many argue that the effective application of OOD calls for the whole software **development** process to be founded upon **object**-oriented concepts. Not surprisingly, progress in CASE tool provision for OOD reflects this uncertainty. Some tool vendors have added a new layer over a traditional model such as entity-relationship (E-R), whereas others have developed totally new tools based entirely upon the **object**-oriented approach. The conceptual basis of conventional methodologies such as SSADM is essentially functional in nature. Systems are analysed and modelled in terms of information flows and processes functional transformations from sets of inputs to sets of output. **Object**-oriented **design** is neither a functional nor a data decomposition process. In contrast, its approach is to identify the real world **objects** that belong to the problem domain, and then classify them. Conventional methodologies require that the **designer** first establish what the system does, whereas OOD tries to avoid this question for as long as possible. Ultimately, this is the key to the OOD strategy for tackling the problems of maintenance and system evolution. In the OOD view, system functionality is far more likely to change over time than are the **objects** from which the system is built. Thus, it is essential to **make** the whole **development** process **object**-oriented. Unfortunately, **object**-oriented methodologies remain uncommitted as to how best to do **object**-oriented software **development**, beyond denouncing the waterfall model and embracing the iterative, evolutionary 'spiral' model. This situation represents perhaps the biggest obstacle to the further uptake of OOD techniques, for although the benefits of the spiral model may be apparent to software engineers, many of those who are charged with the responsibilities of project planning and management have yet to be convinced.

COPYRIGHT 1992 by Elsevier Science Publishers Ltd.

COPYRIGHT 1992 Elsevier Science Publishers Ltd.

COPYRIGHT 1999 Gale Group

PUBLISHER NAME: Elsevier Science, Inc.

INDUSTRY NAMES: BANK (Banking, Finance and Accounting); BUSN (Any type of business); CMPT (Computers and Office Automation); INTL (Business, International)

(USE FORMAT 7 FOR FULLTEXT)

TEXT:

Object -oriented programs consist of a **collection** of interacting **objects** that are instances of classes. **Object** -oriented **design** is based upon two essential concepts: encapsulation and inheritance. Encapsulation refers to the principle of storing both...

...and the process (known as 'methods' in OOD parlance) that act upon them as indivisible entities, or **objects**. In its behaviour, therefore, an **object** is just like a very simple computer. It is capable of acting on **instructions** ('messages'), and of sending them to other **objects**. Messages are the only means of access to the data **associated** with an **object**. They specify what operation is required on the data, but not how it is **implemented**. Both the data and methods of a given **object** are private, and it is this which gives **objects** and their operations their inherent modularity. In many respects, **objects** are simply another instance of the information hiding principles first embodied in the concept of abstract data...

...languages already support the specification of abstract data types, but they lack the concept of inheritance. Each **object** is a member of a specific class, and this defines its **properties**. Classes are related to one **another** through inheritance. Aggregation and generalization are forms of inheritance which permit existing **object** classes to be grouped together to **produce** a new **object** class. For example, the **object** class customer might be defined as an aggregation of the classes current account, deposit account, mortgage etc., whilst the class **employee** might be defined as the generalization of the classes branch manager, teller, ledger clerk etc. Specialization is the reverse process of generalization, **creating** a new class by adding **properties** to an existing one, known as the parent class. Inheritance may be strict, in which case instances of a class must satisfy all the **properties** of its parent. By contrast, default inheritance allows instances of a class to contradict some **properties** of its parent. A further extension is multiple inheritance, in which a new **object** class may be a specialization of more than one parent. OOD typically begins with an informal description of the solution strategy. From this, the next step is to isolate **objects** and operations. **Objects** are identified by the noun **occurrences** within the description, and represent the information domain. Operations, which are identified by verb **occurrences**, represent the processing **associated** with the information domain. The identification of **objects** and their **attributes** is the foundation for the creation of the system's data abstractions. Consideration of **objects** and operations together leads to the specification of the system's structure. The relationships between **objects** and operations leads to the definition of how the **objects** are used, and their interfaces, i.e. the messages that are passed between them. - Impact on Software Development Productivity **Object** -oriented software offers a number of potentially powerful **features** for the **design** of complex systems, and for the management of changes in data, and system functionality, over long periods of time. Proponents of OOD argue that one of the problems with conventional software **design** is that data and processing are regarded as two quite separate perspectives of the **design** task; OOD offers a way to re-integrate them. Encapsulation is a tool with great potential. In theory, it means that it is not necessary to consider other **objects** in the system when considering the new specific **object**. To **use**, test or re- **use** an **object**, it should only be necessary to understand its interface, i.e. the messages that can be sent to that **object**, and what they do. Similarly, when extending the **implementation** of a particular **object**, it should be possible to ignore all other **objects**. The question of why software **production** has failed to follow the **pattern** set by hardware is a vexed one. It has been estimated that only 15% of new **code** actually serves a new purpose; the rest merely duplicates what many others have written before. Software libraries...

...apart from one or two exceptions (e.g. mathematical routines), have made little impact on commercial software **development** practice. In OOD, the problem of re-usability is identified with the lack of abstraction in conventional software **design**. OOD aims to raise the level of abstraction

of software **design** through the exploitation of **object** classes and inheritance. A well **designed** class hierarchy enables the software engineer to **create** new **objects** by building upon the **properties** inherited from an existing class. It is then only necessary to define how a new **object** differs from that class. With OOD techniques, many believe that the goal of building systems from 'software ICs', selected 'off the shelf' from catalogues of **objects**, will finally be achieved. In summary, claims that OOD improves software **productivity** rest on four basic premises: 1. **Objects** and classes reduce the difficulties of translating complex real world situations into systems. In conventional programming languages, the **designer** must map the problem domain onto predefined data and control structures. In OOD, the **designer** can **create objects** and classes which closely match those in the problem domain. It is argued that inheritance is a...

...semantics, and that OOD reduces the conceptual gap between the real world and its computer model. 2. **Object**-oriented **code** is flexible, and so permits rapid responses to changing user requirements. Inheritance implicitly encourages **code re-use**, and so helps economize on **implementation** and maintenance effort. 3. The **re-use** of standard components reduces **development** time and reduces the **code** size. A company can build up libraries of classes tailored to its specific needs and **use** them over and over again. 4. Maintenance costs are reduced. Information hiding means that **design** is decoupled from **implementation**, and changes are effectively isolated. In conventional software, where data and procedures are packaged separately, changes in data structure may affect many different modules, which in turn may have been written by many different **programmers**. In OOD, since data and procedures are packages together, the implications of most changes can ...because they are so localized. The Software Engineering Institute at Carnegie Mellon University has defined five software **development** 'maturity levels', which range from 'chaotic happy hacking' at level one, to the application of a systemic process with continuous self improvement at level five. According to surveys, most software **development** organizations are still below level two on this scale. OOD promises to help improve the maturity level of software **development**. The naturalness of the **object** notion facilitates moving smoothly from requirements via analysis and **design** to an **implementation**. In addition, it is claimed that **object**-oriented analysis yields qualitatively better models than structured analysis modelling. This should result in a deeper understanding...

...environments, the importance and difficulties of integrating with existing systems, the lack of support for large scale **development** both in terms of methodologies and tools and the shortage of appropriate skills. Now, however, some of these obstacles are beginning to diminish. - Practical Experience with **Object**-Oriented **Design** Software engineers are entitled to treat claims that OOD will transform software **development** with some scepticism. There is now enough evidence from experience in the **field**, however, to begin to evaluate their legitimacy. Studies of the impact of **object**-oriented techniques on software maintenance have provided some encouraging results. Compared with a system **implemented** in a conventional programming language, **object**-oriented **code** was found to require fewer module changes, fewer source line changes, and fewer source lines added. Another...

...concluded that OOD may cut maintenance effort by as much as 50%. There is also confirmation for **code** re-usability promoting gains in **productivity**. In some cases, reductions of 5:1 in the amount of **code** required for a new application have been reported, together with a substantial reduction in the size and complexity of the resultant **code**. OOD techniques have been found to raise their own **design** problems, however. A simple example is whether some part of an **object** should be represented as an instance variable, or inherited as a component class. Decisions of this kind...

...may only be arrived at through trial and error, by which time it may be difficult to **implement** changes. The view persists that the key to

maintenance is good **design** rather than the specifics of the tools. A large **object** -oriented system without a proper **design** foundation may be just as unmaintainable as a traditional pile of Fortran or Cobol 'spaghetti'. In commercial environments, the temptation to **produce code** quickly is the same no matter what tools are available. Conceivably, it is easy to **produce** hard to read and hard to maintain **object** -oriented **code**. The longevity of **object** -oriented **code** then reveals its disadvantages - badly written **code** propagates. It has been argued that **features** such as inheritance may actually compromise the requirements of large scale software projects. For example, the integration...

...would be. Opinions differ over whether the best results can be obtained by judiciously mixing OOD with **more** conventional software **development** methodologies. Clearly, anything which might reduce the potential upheaval would be highly desirable. - Integration with Conventional Methodologies...

...achieved in this way; many argue that the effective application of OOD calls for the whole software **development** process to be founded upon **object** -oriented concepts. Not surprisingly, progress in CASE tool provision for OOD reflects this uncertainty. Some tool vendors...
...such as entity-relationship (E-R), whereas others have developed totally new tools based entirely upon the **object** -oriented approach. The conceptual basis of conventional methodologies such as SSADM is essentially functional in nature. Systems...

...in terms of information flows and processes functional transformations from sets of inputs to sets of output. **Object** -oriented **design** is neither a functional nor a data decomposition process. In contrast, its approach is to identify the real world **objects** that belong to the problem domain, and then classify them. Conventional methodologies require that the **designer** first establish what the system does, whereas OOD tries to avoid this question for as long as...

...In the OOD view, system functionality is far more likely to change over time than are the **objects** from which the system is built. Thus, it is essential to **make** the whole **development** process **object** -oriented. Unfortunately, **object** -oriented methodologies remain uncommitted as to how best to do **object** -oriented software **development**, beyond denouncing the waterfall model and embracing the iterative, evolutionary 'spiral' model. This situation represents perhaps the...

25/5,K/19 (Item 2 from file: 636)
DIALOG(R) File 636:Gale Group Newsletter DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01388315 Supplier Number: 41764265 (THIS IS THE FULLTEXT)

THE IBM/TIRS APPROACH

Intelligent Software Strategies, v7, n1, pN/A
Jan, 1991

ISSN: 1052-7214

Language: English Record Type: Fulltext

Document Type: Magazine/Journal; Trade

Word Count: 207

TEXT:

Another example of a knowledge-base **development** methodology that incorporates elements similar to those in Aion's methodology is IBM's Target & DARTS (Data And Rule TemplateS) methodology. IBM positions its Target & DARTS methodology as an analysis and **design** methodology and has developed it as a methodology that is specifically adopted for TIRS **development**. Like Aion's **Development** Methodology, Target & DARTS **uses** worksheets (tables) to **generate objects** (i.e., frames) and it **uses pattern** -matching **rules** to reason about the resulting domain model. Target & DARTS also depends on an analysis of problem types which are divided into three groups. Group one includes: Selection, Filtering, Screening, Scoring, and Advice-Giving. Group 2 includes: Help Desk, Procedures Manual, and Diagnosis. Group 3 includes: Allocation and

Scheduling. IBM is currently offering courses in its Target & DARTS methodology to selected clients. We fully expect to see the other major expert systems tool vendors (as well as the consultants like Andersen) introducing similar approaches in 1991. Moreover, we expect they will be similar to ADM and Target & DARTS simply because the hybrid tools seem to suggest that an effective expert systems methodology will **combine** the same basic set of **features**. (Just as the various structured methodologies developed for **use** with COBOL all have a similar set of **features**.)

COPYRIGHT 1991 Cutter Information Corp.

COPYRIGHT 1999 Gale Group

PUBLISHER NAME: Cutter Information Corp.

COMPANY NAMES: *International Business Machines Corp.

EVENT NAMES: *330 (Product information)

GEOGRAPHIC NAMES: *1USA (United States)

PRODUCT NAMES: *7372005 (Artificial Intelligence Software)

INDUSTRY NAMES: BUSN (Any type of business); CMPT (Computers and Office Automation)

NAICS CODES: 51121 (Software Publishers)

TICKER SYMBOLS: IBM

(USE FORMAT 7 FOR FULLTEXT)

TEXT:

Another example of a knowledge-base **development** methodology that incorporates elements similar to those in Aion's methodology is IBM's Target & DARTS (Data And **Rule** Templates) methodology. IBM positions its Target & DARTS methodology as an analysis and **design** methodology and has developed it as a methodology that is specifically adopted for TIRS **development**. Like Aion's **Development** Methodology, Target & DARTS **uses** worksheets (tables) to **generate objects** (i.e., frames) and it **uses pattern-matching rules** to reason about the resulting domain model. Target & DARTS also depends on an analysis of problem types...

...Target & DARTS simply because the hybrid tools seem to suggest that an effective expert systems methodology will **combine** the same basic set of **features**. (Just as the various structured methodologies developed for **use** with COBOL all have a similar set of **features**.)

25/5,K/20 (Item 3 from file: 636)

DIALOG(R) File 636:Gale Group Newsletter DB(TM)

(c) 2003 The Gale Group. All rts. reserv.

01257550 Supplier Number: 41338260 (THIS IS THE FULLTEXT)

INFORMATION BUILDERS BUILDS STRATEGY FOR THE 1990S ON GAPS LEFT BY IBM'S REPOSITORY...

Computergram International, n1428, pN/A

May 18, 1990

ISSN: 0268-716X

Language: English Record Type: Fulltext

Document Type: Newswire; Trade

Word Count: 545

TEXT:

According to New York City-based Information Builders Inc, the 1990s will be the era of information architecture. In other words information technology users will be working in a multi-vendor environment using common applications and common data, and will require an infrastructure of tools and services to fit in with this environment but which will also be compatible with existing systems as well as with new technology. Information Builders believes that IBM's Systems Application Architecture is the best and most well-defined information architecture for this purpose, but thinks that IBM's Repository probably won't have built-in management and control tools to support multi-vendor environments. Consequently, this is the market that the company intends to go for throughout the decade. This means that its Focus will have to be more than a fourth **generation** language and will evolve into a complete tool for information management. Last month the company made two announcements that indicated that it was beefing up its Focus offering to meet this newly

targeted market: firstly it integrated its Level 5 expert system with the Focus language to enable developers to write applications containing **embedded** knowledge. Secondly, Information Builders' new 5.5 release of PC/Focus started shipping: the thing offers a database management system architecture called **Direct/ Connect** between personal computers and mainframes. This latest release of PC/Focus fulfils Phase 1 of Information Builders' long-term Focnet strategy to **implement** client-server co-operative processing unveiled last year in Cannes (CI No 1,300). Next year Focus procedures will be distributed and stored as executable **objects** and will be accessed via remote procedure calls for cross-machine **object** execution. Phase 3 of the strategy is on schedule for delivery in 1992 when Focus will be a completely interoperable system with applications being interoperable with servers, and will have universal communications support and system-wide data transparency. However, Information Builders' head of marketing John Senor was at pains to explain that this will not be a true distributed database, merely distributed applications. He doesn't think that IBM's distributed database will arrive this century. As for **other developments** about to be launched on an unsuspecting world, Information Builders has just released its Level 5 **Object product**, taking Focus into the graphical user interface environment and offering portability across graphics environments. However, the company's president Gerry Cohen is sceptical about the impact that graphical user interfaces will have, since he thinks that they are too complicated for databases to handle properly. Similarly, he doesn't have too much time for **object**-oriented techniques, saying that while the press and theoreticians love them, he's not so sure about users. He suggests half-teasingly that users can get a "dosage" of such techniques by using Level 5 **Object** which will "tell you more than you ever want to know about using that particular paradigm". Focus will also be available on the AS/400 in June. Looking a little further ahead to the autumn, Information Builders will have Presentation Manager Focus available: this will **use object**-oriented techniques by **applying** the Focus language to discrete screen **objects**, thereby **producing** modular, reusable **code**. At the same time the company will introduce the Focus Application Creation Tool in a version which can be used to **generate** Focus **code** by picture programming. - Katy Ring

COPYRIGHT 1990 by Apt Data Services Ltd.

COPYRIGHT 1990 ComputerWire Inc.

COPYRIGHT 1999 Gale Group

PUBLISHER NAME: ComputerWire, Inc.

INDUSTRY NAMES: CMPT (Computers and Office Automation); INTL (Business, International)

(USE FORMAT 7 FOR FULLTEXT)

TEXT:

...for throughout the decade. This means that its Focus will have to be more than a fourth **generation** language and will evolve into a complete tool for information management. Last month the company made two...

...integrated its Level 5 expert system with the Focus language to enable developers to write applications containing **embedded** knowledge. Secondly, Information Builders' new 5.5 release of PC/Focus started shipping: the thing offers a database management system architecture called **Direct/ Connect** between personal computers and mainframes. This latest release of PC/Focus fulfils Phase 1 of Information Builders' long-term Focnet strategy to **implement** client-server co-operative processing unveiled last year in Cannes (CI No 1,300). Next year Focus procedures will be distributed and stored as executable **objects** and will be accessed via remote procedure calls for cross-machine **object** execution. Phase 3 of the strategy is on schedule for delivery in 1992 when Focus will be...

...distributed applications. He doesn't think that IBM's distributed database will arrive this century. As for **other developments** about to be launched on an unsuspecting world, Information Builders has just released its Level 5 **Object product**, taking Focus into the graphical user interface environment and offering portability across graphics environments. However, the company...

"
...are too complicated for databases to handle properly. Similarly, he doesn't have too much time for **object** -oriented techniques, saying that while the press and theoreticians love them, he's not so sure about...

...He suggests half-teasingly that users can get a "dosage" of such techniques by using Level 5 **Object** which will "tell you more than you ever want to know about using that particular paradigm". Focus...

...a little further ahead to the autumn, Information Builders will have Presentation Manager Focus available: this will **use object** -oriented techniques by **applying** the Focus language to discrete screen **objects** , thereby **producing** modular, reusable **code** . At the same time the company will introduce the Focus Application Creation Tool in a version which can be used to **generate** Focus **code** by picture programming. - Katy Ring